НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет електроніки
(повна назва інституту/факультету)

Кафедра звукотехніки та реєстрації інформації
(повна назва кафедри)

«На правах рукопису»
УДК 621.397.63

«До захисту допущено»

Завідувач кафедри
_____ Г.Г. Власюк
(підпис) (ініціали, прізвище)

“_9_”___грудня___2019 р.

Магістерська дисертація

спеціальність 171 «Електроніка»
(код і назва)

на тему: «Забезпечення безпеки в системах пристроїв Інтернету речей»

Виконав: студент II курсу, групи ДВ-82мп
(шифр групи)

Лихно Руслан Станіславович _____
(прізвище, ім’я, по батькові) (підпис)

Керівник к.т.н., доцент, Попович П. В. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____ _____
(науковий ступінь, вчене звання, , прізвище, ініціали) (підпис)

Рецензент _____ _____
(посада,науковий ступінь,вчене звання,науковий ступінь,прізвище,ініціали) (підпис)

Засвідчую, що у цьому дипломному проекті немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2019 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет електроніки

Кафедра звукотехніки та реєстрації інформації

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою

Спеціальність (освітня програма) 171 «Електроніка» («Електронні системи мультимедіа та засоби Інтернету речей»)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Г.Г. Власюк
(підпис) (ініціали, прізвище)
«26» вересня 2018 р.

ЗАВДАННЯ
на магістерську дисертацію студенту

Лихну Руслану Станіславовичу
(прізвище, ім'я, по батькові)

1 Тема роботи «Забезпечення безпеки в системах пристроїв Інтернету речей»
керівник роботи Попович П. В., к.т.н., доцент.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «07» листопада 2019р. №3859-с

2 Строк подання студентом дисертації 9 грудня 2019 р.
3. Об'єкт дослідження технології забезпечення безпеки систем Інтернету речей в процесі інтеграції

4. Предмет дослідження (Вихідні дані – для магістерської дисертації за освітньо-професійною програмою) вплив параметрів безпеки систем IoT на стан, ефективність та економічну доцільність підприємств різних галузей

5. Перелік завдань, які потрібно розробити: проаналізувати існуючі на даний момент вразливі та слабкі місця в системах IoT, види та типи атак, запропонувати технології, що дозволяють вирішити питання безпеки в системах Інтернету речей,

дослідити та порівняти інтеграції кінцевих пристроїв IoT з опорною мережею через використання мережних сервісів

6. Перелік графічного (ілюстративного) матеріалу 12 слайдів презентації: характеристика роботи, формулювання завдання роботи, загальні характеристики та типи вразливостей, існуючі рішення забезпечення безпеки, технології, що дозволяють покращити рівень безпеки систем Інтернету речей при мінімальних витратах, аналіз і порівняння результатів дослідження реалізації включення кінцевого IoT пристрою через мережний сервіс, висновки

7. Орієнтовний перелік публікацій:

1) Спосіб вибору точки збору інформації в мобільних сенсорних мережах / Лихно Р. С. // II Всеукраїнська науково-технічна конференція «Сучасні технології кіно та аудіовізуальних систем - 2019» 2) Підвищення безпеки та пропускної спроможності мережі IoT / Лихно Р. С. // II Всеукраїнська науково-технічна конференція «Сучасні технології кіно та аудіовізуальних систем - 2019»

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання 26 вересня 2018 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Написання першого розділу	15.03.2019	
2	Написання другого розділу	19.06.2019	
3	Написання третього розділу	15.10.2019	
4	Написання четвертого розділу	12.11.2019	
	Написання п’ятого розділу	27.11.2019	
5	Підготовка матеріалів до друку та оформлення пояснювальної записки	01.12.2019	
6	Підготовка та оформлення презентації для доповіді	10.12.2019	

Студент _____ Р. С. Лихно
(підпис) (ініціали, прізвище)

Керівник роботи _____ П. В. Попович
(підпис) (ініціали, прізвище)

УДК 621.397.63

РЕФЕРАТ

Магістерська дисертація: 104 с., 18 рис., 28 табл., 1 дод., 20 джерел

БЕЗПРОВОДОВА МЕРЕЖА, WI-FI, 3G, TLS, ІНТЕРНЕТ РЕЧЕЙ, ПЕРЕДАВАННЯ ДАНИХ, SSL, МЕРЕЖНІ СЕРВІСИ, ЯКІСТЬ, БЕЗПЕКА, MQTT, АВТЕНТИФІКАЦІЯ, AMAZON AWS, DDOS, ПОЛІТИКА УПРАВЛІННЯ

Актуальність роботи. З кожним роком з появою нових систем, знаходяться вразливі місця, які ставлять під питання безпеку систем в цілому. Особливо важливу роль починає займати Інтернет речей, який стрімко охоплює все більше аспектів сучасного життя, завдяки технологічному прогресу в галузі безпроводових технологій як Wi-Fi, ZigBee, LTE. Оскільки рішення на базі IoT використовуються не тільки в домашніх мережах, але й на фабриках, підприємствах, великих компаніях та державних установах, особливо гостро постає питання забезпечення безпеки в таких системах, враховуючи цінність даних, які вони опрацьовують та пропускають крізь себе, та масштаб наслідків які можуть виникнути внаслідок втрати даних або захоплення контролю над системою третіх осіб.

Основною проблемою є відсутність комплексного захисту кожного найменшого елементу системи IoT, оскільки навіть один пристрій з підключенням до інтернету дає змогу зловмисникам отримати доступ на програмному чи апаратному рівні до всієї системи.

Мета і завдання дослідження. *Метою* роботи є підвищення рівня безпеки в системах Інтернету речей для комплексного їх захисту в залежності від особливостей систем та наслідків, до яких може привести наявність вразливих місць.

Для досягнення поставленої мети необхідно вирішити такі *завдання*:

– проаналізувати основні вразливості систем Інтернету речей з огляду на потенційні загрози;

– дослідити існуючі рішення для захисту та моніторингу стану таких систем;

– запропонувати технології, що дозволяють значно підвищити рівень безпеки в системах IoT та надають можливість гнучкого і масштабованого реагування в майбутньому для боротьби з новими загрозами, які змінюються щодня;

– експериментально дослідити підключення пристрою IoT через мережний сервіс AWS для швидкої та ефективної реалізації комплексу засобів безпеки.

Об'єкт дослідження – технології забезпечення безпеки та комплексні рішення для систем Інтернету речей.

Предмет дослідження – загальне оцінювання актуальних загроз та систем захисту, які існують і використовуються на сьогодні, та перспективних технологій, які можуть надати високий рівень захисту і будуть на крок попереду від нових алгоритмів атак на системи Інтернету речей.

Методи дослідження – теоретичне дослідження систем захисту та основних вразливих місць в різноманітних втіленнях систем Інтернету речей, порівняльний аналіз різних реалізацій на основі мережних рішень.

Наукова новизна одержаних результатів. Запропоновано використовувати мережні сервіси для якісного покращення рівня безпеки, масштабованості та гнучкості в системах Інтернету речей та моніторингу всіх основних показників безпеки, що дозволяє адаптувати безпеку під більшість системних рішень незалежно від бюджету та складності поставлених завдань.

Практичне значення одержаних результатів. Розроблений макет системної реалізації кінцевого пристрою IoT з підключенням до мережного сервісу AWS має істотні переваги в плані безпеки, управління, моніторингу та надає доступ до великої кількості інших сервісів платформи Amazon, що можуть бути використані для оптимізації та покращення систем, в рамках яких буде інтегровано дане рішення.

Апробація результатів дисертації. Результати досліджень, що включені до дисертації, оприлюднені на II Всеукраїнській науково-технічній конференції «Сучасні технології кіно- та аудіовізуальних систем» (2019).

Публікації. Результати досліджень, наведених в дисертації, оприлюднено в таких виданнях:

1. Р. С. Лихно. Спосіб вибору точки збору інформації в мобільних сенсорних мережах / П.В. Попович, Р.С. Лихно// Матеріали конференції «Проектування та оптимізація інформаційних та телекомунікаційних систем». – К: КПІ ім. Ігоря Сікорського, 2019. - С. 21.

2. Р. С. Лихно. Підвищення безпеки та пропускної спроможності мережі IoT / П.В. Попович, Р.С. Лихно// Матеріали конференції «Захист, безпека та електромагнітна сумісність». – К.: КПІ ім. Ігоря Сікорського, 2019. - С. 86.

SUMMARY

The subject of IoT security, then, is not the application of a single, static set of metasecurity rules as they apply to networked devices and hosts. It requires a unique application for each system and system-of-systems in which IoT devices participate. IoT devices have many different embodiments, but collectively, an IoT device is almost anything possessing the following properties

The result of this work is the proposal to use cloud services to improve the security, scalability and flexibility of the Internet of Things systems, and to monitor all key metrics. The adaptability of this solution to most system solutions, regardless of budget and complexity of tasks, when implemented in the short term.

Research data on system implementation of an IoT terminal device with connection to a cloud service can be used in various variants of implementation of the Internet of Things systems, which is now common in almost all spheres of activity of companies and our life as a whole.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

CAN – Controller Area Network (контролер мережі);

CPS – Cyber-physical Systems (кібер-фізична система);

CSP – Cloudvisory Security Platform (хмарна платформа безпеки);

DANE – DNS-based Authentication of Named Entities (аутентифікація по доменним іменам);

DoS – Denial of Service (відмова в обслуговуванні);

DSRC – Dedicated short-range communications (радіозв'язок близькьої дії в транспортному середовищі);

ECU – Electronic Control Unit (електронний блок управління);

EEPROM – Electrically Erasable Programmable Read-Only Memory (постійний запам'ятовувальний пристрій);

FPGA – Field-Programmable Gate Array (Програмована користувачем вентильна матриця);

HSM – Hardware Security Module (блок фізичної безпеки);

MCU – Micro Controller Unit (мікроконтролер);

MQTT – Message Queue Telemetry Transport (транспортний протокол);

REST – Representational State Transfer (передача репрезентативного стану);

RTOS – Real-time operating system (операційна система реального часу).

SDN – Software Defined Networking (програмно визначена мережа);

TLS – Transport Layer Security (протокол безпеки транспортного рівня);

URL – Uniform Resource Locator (єдиний вказівник на ресурс);

WSN – Wireless sensor network (безпроводова сенсорна мережа);

ЗМІСТ

ВСТУП

Інтернет речей (Internet of Things, IoT) – значний крок в прогресі засобів зв'язку, який все ще набирає обертів. Різноманіття варіацій застосування, функцій, його вплив на життя людей, комерційну діяльність важко оцінити та спрогнозувати майбутнє. IoT визначає комплекс підключених один до одного пристроїв, від побутової електроніки до сенсорів та датчиків. Сьогодні Інтернет реалізує з'єднання між собою незліченної кількості промислових і побутових предметів, як правило, за допомогою хмарних систем. Такі предмети передають інформацію про оточення, аналізуючи та передаючи дані з датчиків сотні разів в секунду в залежності від призначення.

Інтернет речей все змінює. На жаль, у багатьох галузях промисловості та комерційних реалізаціях пристрої та інфраструктурні елементи швидко проявляють свої вразливі місця, що може загрожувати серйозними наслідками. Прагнення зробити всі пристрої «розумними» надає багато нових можливостей кіберзлочинцям. Ці загрози будуть лише зростати, якщо вони потенційно впливатимуть на економіку, корпорації, господарські операції, приватне життя та безпеку. Багато профільних компаній в галузі досліджень безпеки надають зовсім невтішні прогнози щодо основних вразливостей та порушень безпеки в усіх сенсах. Деякі з цих порушень можуть призвести до руйнації компаній, а головне – можуть завдати значної шкоди окремим громадянам.

Історично багато з галузей та пристрої, такі як розумні холодильники та пральні машини з операційними системами, які на разі охоплені Інтернетом речей, ніколи не мали б стосуватися кібербезпеки в її класичному визначенні. Однак, зважаючи на гарячу конкуренцію нових товарів та продуктів, вони тепер опиняються на небезпечній території, де невідомо, як розвивати, розгортати та інтегрувати свої пристрої в комплексі IoT, задовольнивши всі норми безпеки.

1 АНАЛІЗ ВРАЗЛИВОСТЕЙ ТА АТАК НА СИСТЕМИ ІНТЕРНЕТУ РЕЧЕЙ

Майже неможливо обговорити практичні аспекти загроз, вразливостей та ризиків без визначення істотних компонентів забезпечення інформаційної безпеки, що є важливим елементом безпеки IoT систем:

– *конфіденційність*: зберігання конфіденційної інформації в таємному та захищеному від розкриття стані;

– *цілісність*: забезпечення того, що інформація не буде змінена випадково або цілеспрямовано, без відстеження даних дій;

– *перевірка автентичності*: забезпечення того, що джерело даних має визначене джерело або кінцеву точку;

– *безвідмовність*: гарантування того, що людина чи система не зможуть завадити виконанню функцій;

– *доступність*: забезпечення доступності інформації при необхідності.

Задоволення вимог інформаційної безпеки не обов'язково означає, що організація повинна виконувати всі попередні пункти на практиці. Не всі дані, наприклад, потребують конфіденційності. Інформація та категоризація даних – це складна тема сама по собі і не вся інформація є критичною або важливою. Для правильного моделювання загрози пристроям, додаткам та даним, які вони включають, потрібна структуризація для виявлення чутливості як окремих елементів даних, так і даних у загальній формі. Ризики агрегації великих, здавалося б, доброякісних наборів даних Інтернету речей є одним з найскладніших викликів. Добре визначені категорії даних і комбінаційні обмеження дозволяють мати певні гарантії, такі як конфіденційність або цілісність, які визначенені для кожного елемента даних або складного типу інформації.

1.1 Загрози

Важливо розрізняти загрозу та джерело загрози (або суб'єкт загрози). Кожна загроза має джерело впливу. Наприклад, у випадку вторгнення у ваш будинок, прийнято вважати грабіжника фактичною загрозою, але більш точно і логічно вважати його джерелом загрози (виконавцем). Він суб’єкт, який може напасти на ваш будинок для різних шкідливих цілей, особливо це стосується його бажання відокремити ваші цінні активи. У цьому контексті загрозою є потенційний грабіж.

В загальному загрози можуть бути різного типу, як природними, так і техногенними. Торнадо, повені та урагани – можна вважати природними загрозами; у цих випадках погода на Землі служить діючою загрозою (або діями Божими на мові багатьох страхових полісів).

Узагальнений набір загроз для Інтернету речей включає все, що пов’язано з інформаційною безпекою для менеджменту та даних програм, що надсилаються на пристрої IoT та з них. Крім того, пристрої IoT мають такі вразливі місця: фізична безпека, апаратне забезпечення, якість програмного забезпечення, ланцюг передавання даних та багато інших.

CPS (Cyber-Physical Systems – кібер-фізичні системи) піддаються впливу фізичним загрозам щодо надійності, стійкості та виходять за рамки компромісу нормальної роботи та деградації обчислювальної платформи. Важливу роль у CPS грають додаткові інженерні дисципліни, такі як класична теорія управління, оцінка стану та інші, які використовують датчики, сенсорний зворотний зв'язок, контролери, фільтри та приводи, що управляють станом фізичної системи. Загрози можуть також орієнтуватися на функції передавання системи управління, стан та фільтри оцінювання (наприклад, фільтри Калмана) та інші артефакти внутрішнього циклу управління, що мають пряму реакцію та наслідки у фізичному світі.

1.2 Вразливості

Вразливість – це термін, який ми використовуємо для виявлення слабкості в дизайні чи архітектурі, інтеграції або функціонуванні системи або пристрою. Вразливості існують постійно і незліченну кількість відкривають щодня. Безліч баз даних та Інтернет в цілому тепер є порталами які надають нам автоматизовані оновлення щодо нововиявлених вразливостей. На схемі нижче наведено взаємозв'язок між кожним із цих ключових термінів (рис. 1.1).

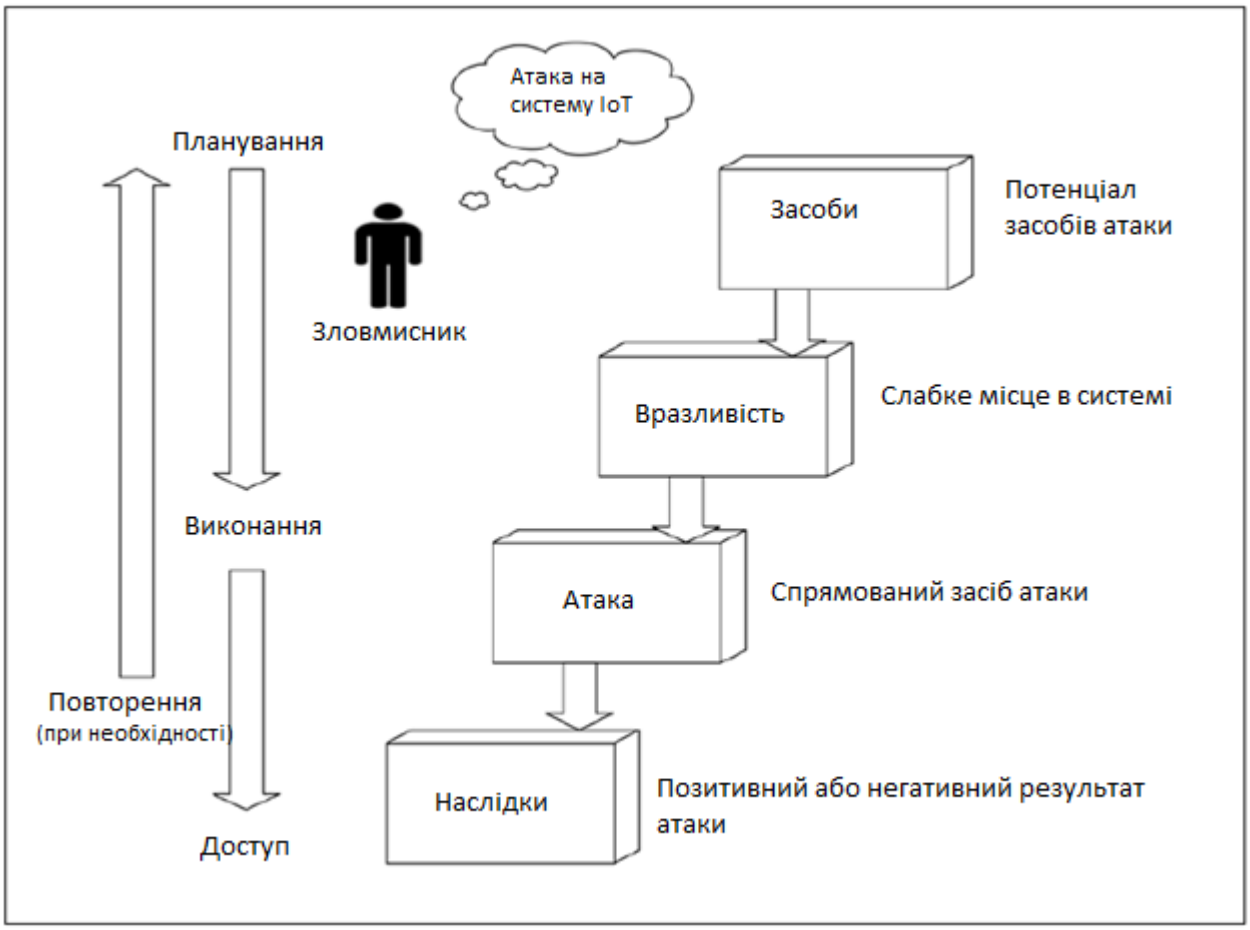

Рисунок 1.1 – Схема впливу на вразливі місця в системах IoT

Вразливості можуть бути недоліками у фізичному захисті пристрою (наприклад, слабкі місця в корпусі пристрою), у якості програмного забезпечення, конфігураціях, відповідності протоколам захисту чи доцільності самих протоколів. Вони можуть містити майже все, що є в пристрої, від недоліків впровадження дизайну

в апаратному забезпеченні (наприклад, дозволяючи підробку FPGA – Field-Programmable Gate Array або EEPROM – Electrically Erasable Programmable Read-Only Memory) до внутрішньої фізичної архітектури та інтерфейсів операційної системи або програм. Зловмисники добре розуміють потенційні вразливості. Вони, як правило, прагнуть виявити найпростіші в плані доступності слабкі місця, найменш затратні або найшвидші в експлуатації.

1.3 Ризики

Можна використовувати якісні чи кількісні методи оцінки ризику. Простіше кажучи, ризик – це можливість нанесення збитків. Він відрізняється від вразливості, бо залежить від ймовірності певної події, нападу чи стану та має міцний зв’язок з мотивацією нападника. Це також залежить від того, на скільки великий вплив має одинична або масові атаки на стан компанії.

Вразливість безпосередньо не викликає подій чи ймовірності, але є слабкістю системи. Спроба вторгнення може бути легкою або важкою, або це призведе до невеликої, або великої втрати в результаті. Наприклад, настільна операційна система може мати серйозний характер вразливості в логіці ізоляції процесу, що дозволяє отримати доступ до ненадійного процесу віртуальної пам'яті іншого додатку. Ця вразливість може бути корисною і, безумовно, являє собою слабку сторону, але якщо система знаходиться локально і ніколи не буде підключена безпосередньо чи опосередковано до Інтернету, вразливість може бути мало ймовірною, інша справа ризик, який є в будь-якому випадку гарантовано небезпечним. Якщо, з іншого боку, платформа підключена до Інтернету, рівень ризику може значно зрости через те, що зловмисник знайде практичний засіб для інтегрування ворожої програмної оболонки, яка використовує вразливість процесу ізоляції і дозволяє зловмиснику взяти на себе право власності над машиною.

Керувати ризиком можна за допомогою моделювання загроз, що допомагає встановити наступне:

– вплив та загальну вартість компромісу;

– наскільки цінні можуть бути цілі для нападників;

– передбачувана майстерність та мотивація нападників (на основі моделювання загроз);

– фундаментальні знання про вразливості системи (наприклад, ті, що є вразливими, виявлені під час моделювання загроз, публічних консультацій, проникнення і так далі).

Управління ризиками покладається на розумне застосування пом'якшення сценаріїв дії на вразливі місця, які, як відомо, є і на які може бути націлена потенційна атака. Закономірним є те, що не всі вразливості будуть відомі до цього часу. Ми знаємо, що певні вразливості є в нашій операційній системі Windows; отже, ми застосовуємо добре підібрані антивірусні програми, мережеве обладнання для моніторингу та зменшення ризиків. Тому що пом’якшення та контроль безпеки ніколи не є ідеальним, у нас все ще залишається невеликий рівень ризику, який зазвичай називають залишковим ризиком. Залишковий ризик часто приймається таким, який є, або компенсується застосуванням інших механізмів компенсації ризику, таких як страхування.

1.4 Поширені типи атак на системи IoT

Існує багато типів нападу; однак, наступний перелік описує кілька найбільш значущих, оскільки вони стосуються Інтернету речей:

– проводове та безпроводове сканування атак;

– протокольні атаки;

– атаки підслуховування (втрата конфіденційності);

– криптографічний алгоритм та атаки на керування ключами;

– підробка і маскування (атаки аутентифікації);

– атаки на цілісність операційної системи та додатків;

– відмова в обслуговуванні;

– фізичні атаки безпеки (наприклад, підробка, симуляція інтерфейсу);

– атаки контролю доступу (ескалація привілеїв).

Згадані вище напади – лише невеликий зразок того, що існує в наш час. У реальному світі, однак, більшість атак налаштовані на конкретні, відомі вразливості. Вразливість, яка ще не є загальновідомою і для якої експлойт (Експлойт (від англ. exploit — експлуатувати) — це комп'ютерна програма, фрагмент програмного коду або послідовність команд, що використовують вразливості в програмному забезпеченні та призначені для проведення атаки на обчислювальну систему), як правило, розроблений, називається вразливістю нульової доби (або O-day). Добре реалізований контроль безпеки має життєво важливе значення для зниження ймовірності або тяжкості наслідків після атаки.

Типи та кількість атак на системи IoT будуть зростати з часом, а в деяких випадках будуть дотримуватися тенденцій мотивації прибутком, подібних до тих, що ми бачимо в кібербезпеці, що розвивається в промисловості. Наприклад, сьогодні спостерігається тривожна тенденція в бізнесі зі зловмисними програмами внаслідок чого зловмисники використовують криптографічні алгоритми для шифрування особистих даних жертви, даних жорсткого диску. Потім зловмисники пропонують повернути дані, розшифровані, за певну плату. Можливий потенціал для такої атаки у царині IoT лякає. Подумайте про зловмисного хакера, який здійснює напади з метою викупу на фізичну інфраструктуру або медичне обладнання. Хтось отримує сповіщення, що серцевий кардіостимулятор був несвідомо компроментований, потерпілий отримає короткий, смертельний для серця поштовх, а щоб зупинити це, потрібно негайно переказувати кошти на цільовий рахунок або ризикувати здоров’ям, з потенційно летальним кінцем. Розгляньте автомобілі, двері гаражних дверей, які відкриваються та іншу потенційну діяльність, яку зловмисники можуть використовувати для викупу. Інтернет речей повинен сприймати ці напади серйозно і не відхиляти їх як лише потенційно можливі. Найбільшою проблемою в галузі безпеки є пошук методів захисту сьогодні, які будуть захищати від завтрашніх атак.

1.4.1 Атаки які існують в системах Інтернету речей на сьогодні

Багато сьогоднішніх атак на пристрої IoT кінцевих користувачів в переважній більшості проводилися дослідниками з метою покращення стану безпеки Інтернету речей. Ці напади часто привертають широку увагу, і часто призводять до зміни позиції безпеки щодо приладу, який тестується. Такого роду тестування є цінним інструментом, оскільки допомагає виробникам вирішити і усунути вразливості перш ніж пристрій буде прийнятим в експлуатацію широким колом людей та особами з менш доброзичливими мотивами. Однак це, як правило, не приємні новини для виробників. Багато виробників не визначені, з тим, як правильно реагувати на повідомлення про вразливі місця в безпеці від дослідників. Деякі організації активно залучають допомогу дослідницької спільноти через такі організації, як BuildItSecure.ly, де волонтери зосереджуються на виявленні вразливих версій програмного чи апаратного забезпечення за запитом самих розробників. Деякі організації оперують власними департаментами з тестування, в яких фахівці з безпеки знаходять та документують вразливості. Інші організації, намагаються закривати очі на вразливості, про які повідомляється в їхніх продуктах, або, що ще гірше, намагаються притягнути до кримінальної відповідальності дослідників.

Атака, що привернула багато уваги, – це віддалений несанкціонований контроль Джипа Черокі 2014 року дослідниками Чарлі Міллером та Крісом Валасеком у 2015 році. Два відкриття дослідників були дуже добре деталізовані у їхньому звіті «Віддалена експлуатація пасажирського транспортного засобу».

Їхня атака базувалася на значній кількості, перевірених на виробництві слабких ділянок у підключених модулях транспортних засобв. Це дослідження збільшувалося з плином часу, супроводжуючи підвищення кваліфікації в університеті Сан-Дієго, Каліфорнія (UCSD). Експлуатація Джипа спиралася на цілий ряд факторів, які були виявленні вченими, що досліджували дистанційний контроль за транспортним засобом. Автомобільні транспортні засоби використовують шину контролерів (CAN

– Controller Area Network) для управління окремими компонентами, відомі як електронні блоки управління (ECU – Electronic Control Unit) для зв'язку (рис. 1.2).

Рисунок 1.2 – Електронний блок управління Jeep Cherokee

Пристрій ECU фактично критично важливий в плані захисту, використовуючи такі системи, як гідропідсилювач керма. Шина може використовуватись, як правило, для того, щоб підтвердити, що повідомлення, яке передається в шині, надходить з дозволеного джерела та не було підмінено до досягнення адреси призначення.

Для віддаленої експлуатації автомобіля доктор Міллер та містер Валасек скористалися низкою недоліків в інфраструктурі, а також окремих підкомпонентів Джипа. Для початку мобільна мережа підтримувала телематику для автомобіля, де дозволено прямий зв'язок між пристроєм та пристроєм. Це дало змогу дослідникам спілкуватися безпосередньо з транспортним засобом і навіть сканувати потенційних жертв через мережу.

Після підключення до Джипа, дослідники почали братися за інші вади в системі безпеки. Одним із прикладів була особливість, яка була вбудована в мультимедійну систему. Ця особливість була функцією виконання в коді, яка

може бути використана для того, щоб передавати довільні дані. Звідти ще один недолік безпеки: можливість вільно переходити по системі та фактично передавати повідомлення віддалено на шини CAN. В архітектурі Джипу обидві шини CAN були підключені до мультимедійної системи, яка спілкувалася через чіп, що дозволяв оновлення його мікропрограмного забезпечення без криптографічного захисту (наприклад, цифрового підпису). Цей остаточний недолік та отриманий комплекс вразливостей в системах зв'язку автомобіля представлені як невеликі недоліки в багатьох системах, але іноді вони перетворюються у велику проблему, яка може призвести до серйозних, іноді навіть смертельних, наслідків.

1.4.2 Безпроводова розвідка та зчитування

Більшість пристроїв IoT на ринку використовують протоколи безпроводового зв'язку, такі як: ZigBee, ZWave, Bluetooth-LE, WiFi 802.11 та інші. Схожий на процес набору номера, коли хакери сканували через телефонні мережі комутації електронні модеми, сьогодні успішно демонструють дослідники по сканувальним атакам на пристрої IoT. Одним із прикладів є техаська компанія Pretorian, розташована в Остіні, штат Техас, яка використала безпілотник, оснащений сканером протоколів ZigBee для виявлення тисяч приладів IoT з підтримкою ZigBee. Так само, як і звичайне сканування мережі за допомогою таких інструментів, як Nmap, що використовується хакерами для збору інформації про хости, підмережі, порти та протоколи у мережах, застосовуються також проти пристроїв IoT, що може дати контроль над такими речами як: відкриття дверей гаража, блокування вхідних дверей, вмикання та вимикання ліхтарів, тощо. Безпроводова розвідка часто передує повномасштабним атакам на пристрої [9].

1.4.3 Атаки засобами протоколу безпеки

Багато протоколів безпеки можуть запобігати атакам на вразливі місця на етапах проектування (специфікації), реалізації та навіть конфігурації
(у якому встановлені різні, конфігураційні параметри протоколу). Як приклад, дослідниками виявлено під час тестування реалізації на основі ZigBee, що була розроблена для зручного налаштування та використання, але не мала можливості конфігурації для безпеки. Ці процедури дозволили третім особам зчитати обмінюваний мережевий ключ під час утворення пари ZigBee та дало змогу отримати контроль над пристроєм ZigBee.

Для забезпечення безпеки системи необхідно встановити та налаштувати елементи управління безпекою [8].

1.4.4 Фізичні атаки на системи IoT

Фізична безпека – це тема, яку часто не помічають постачальники IoT, які є лише історично знайомі з проектуванням обладнання, приладів та інших інструментів, що не підлягають експлуатації. Фізичні атаки безпеки включають ті, в яких зловмисники фізично проникають у корпус хоста, вбудованого пристрою чи іншого типу обчислювальної платформи IoT для отримання доступу до його процесора, пристроїв пам'яті, та інших чутливих компонентів. Після доступу до відкритого інтерфейсу (наприклад, JTAG – IEEE 1149.1, Standard Test Access Port and Boundary-Scan Architecture), зловмисник може легко отримати доступ до пам'яті, чутливого ключового матеріалу, паролів, даних конфігурації та безлічі інших чутливих параметрів. Багато сьогоднішніх засобів безпеки тепер містять широкий захист від фізичних атак на безпеку. Різні засоби контролю щодо несанкціонованого впливу, механізми реагування на несанкціоновані дії (наприклад, автоматичне стирання пам'яті) та інші методи захисту пристроїв від фізичного проникнення. Чіпи смарт-карт, апаратні модулі безпеки (HSM – Hardware Security Module), та багато інших

типів криптографічних модулів використовують такі засоби для захисту криптографічні змінних, а отже, ідентифікацію пристрою та дані – від компроментації.

1.4.5 Атаки на захист програм

Пристрої та з'єднання IoT можна компроментувати за допомогою атак на додатки або кінцеві точки. Кінцеві точки на програмному рівні включають веб-сервери, а також мобільні пристрої та додатки (наприклад, iPhone, Android), які відіграють роль у контролі над пристроєм. Код програми, що працює на самому пристрої, також може бути безпосередньо під прицілом зловмисників.

Захоплені додатки можуть допомогти знайти способи компрометації хоста програми та прийняти контроль над процесами. Крім того, зворотна інженерія та інші помітні атаки можуть виявити негативні, але все ще поширені вразливості при впровадженні, такі як жорсткі коди ключів, паролі та інші рядки в бінарному коді. Ці параметри можуть бути вразливими у різних атаках.

1.5 Модель загроз для систем IoT

Microsoft в свою чергу визначає продуманий підхід до моделювання загроз, використовуючи кілька кроків для визначення ступеня небезпеки, введеної новою системою. Зауважте, що моделювання загроз – це більш фундаментальний підхід до виявлення загроз та джерел загрози. Моделювання, описане раніше, орієнтоване на нападника і розроблене для показу нюансів, як вразливості можуть бути використані.

Щоб проілюструвати процес моделювання загроз, ми оцінимо загрози для розумної паркувальної системи. Розумна система паркування – це корисна система обліку IoT для аналізу, оскільки вона передбачає розгортання елементів IoT у середовищі з високою небезпекою (дехто може обдурити систему оплати за паркування, якщо вони матимуть нагоду). Система містить декілька кінцевих точок,

які фіксують та подають дані в бекенд інфраструктуру для обробки (рис. 1.3). Система забезпечує аналітику даних для визначення тенденції аналізу для осіб, які приймають рішення, співвіднесення сенсорних даних для виявлення порушників паркування в режимі реального часу та відкриває API для смартфонів, що підтримують для клієнта такі функції, як статус місця для паркування в режимі реального часу та платежі.

Рисунок 1.3 – Макет розумної системи паркування

Багато систем IoT мають архітектуру з подібними компонентами та інтерфейсами. У цьому прикладі наша розумна система паркування відрізняється від рішень для паркування у реальному житті. Наша прикладна система забезпечує більш багатий набір функцій який відображено в наступних пунктах:

– **послуга, спрямована на споживачів**: це дозволяє клієнтам визначати статус послуги і ціни на місця для паркування поблизу;

– **гнучкість оплати**: можливість приймати різні форми оплати, включаючи кредитні картки, готівку / монети та мобільні платіжні послуги (для, наприклад, Apple Pay, Google Wallet);

– **моніторинг стану послуги**: можливість відстежувати виділений час, що купується для місця, коли термін дії права на паркування закінчився, або коли транспортний засіб перевищив придбаний період, повідомляє про порушення використання паркування;

– **тенденційний аналіз**: можливість збирати та аналізувати статистичні дані про паркування та надавати звіти про тенденції керівникам парковки;

– **ціноутворення у відповідь на попит**: можливість змінювати ціни залежно від попиту на кожну зону [7].

Враховуючи, що система призначена для збору платежів від споживачів, попередження правоохоронних органів, коли відбулася несплата та надання послуг ціноутворення на основі поточного попиту на паркування, для досягнення безпеки всіх аспектів даних, структура захисту системи може бути викладена так:

– необхідність підтримки цілісності усіх даних, зібраних в системі;

– збереження конфіденційності даних у системі;

– підтримка доступності системи в цілому та її окремих компонентів.

У межах системи інтелектуального паркування чутливими даними можуть бути визначені дані про оплату, банківські реквізити, які можуть містять конфіденційну інформацію, а також відеозаписи, які захоплюють інформацію номерних знаків.

Висновки до розділу

У цій главі досліджено вразливості систем Інтернету речей та основні види атак, що ілюструють, як організація може практично визначити, охарактеризувати та змоделювати системні загрози для Інтернету речей.

1. Більшість проблем, які викликають занепокоєння – це атаки на сервери, робочі станції та смартфони, слабка автентифікація, незашифровані повідомлення, що надсилаються між пристроями, бази даних SQL та відсутність належного контролю оновлення програмного забезпечення та політик безпеки.

2. Окрім втрати конфіденційності в традиційних мережах зв'язку (підслуховування, викривлення інформації), виникають проблеми із захистом користувацької складової. Вони обумовлені:

– відсутністю стандартів не тільки захисту, але і взаємодії;

– відсутністю в наші дні інтересу у виробників, як у першого ступеня реалізації.

3. Велику загрозу несуть атаки на управління пристроїв за допомогою міжмашинної взаємодії, DDoS-атаки, атаки на інфраструктуру передавання даних та ще багато типових «Man in the middle» атак.

Однак багато пристроїв IoT мають суворі експлуатаційні обмеження в залежності від типу та умов використання. Ці обмеження часто не дозволяють безпосередньо використовувати основні заходи безпеки, такі як реалізація брандмауерів або використання сильних криптосистем для шифрування зв'язку з іншими пристроями, а низька ціна та вузьконаправленість багатьох пристроїв приводить до того, що виробники дуже рідко використовують надійні системи захисту.

2 КОМПЛЕКСНИЙ ПІДХІД ДО АНАЛІЗУ БЕЗПЕКИ СИСТЕМ ІНТЕРНЕТУ РЕЧЕЙ

Великі організації стикаються з проблемою розгортання не тільки тисяч пристроїв в одній системі IoT, але й потенційно сотнями чи тисячами індивідуальних кінцевих точок IoT. Збільшуючи складність, кожна реалізація IoT може суттєво відрізнятися за формою та функціями. Наприклад, організація, яка працює в магазинах роздрібної торгівлі, може мати складові RFID-системи, що використовуються в управлінні запасами, маячки в торгових закладах, які підтримують накопичення індивідуальних даних клієнтів, а також можуть почати використовувати технології, такі як підключені транспортні засоби, безпілотники та робототехніку в різних аспектах їхніх операцій.

Завдання інженера з безпеки полягає в тому, щоб мати можливість вивчити та охарактеризувати кожну з цих розрізнених систем та визначити відповідний життєвий цикл, орієнтований на підтримку безпечного стану на підприємстві. У цьому розділі розглядається життєвий цикл безпеки систем Інтернету речей, який тісно інтегрований у процес розробки, інтеграції та розгортання. Життєвий цикл розроблений так, щоб він був ітеративним, що дозволяє забезпечити безпечне додавання нових можливостей IoT на всьому підприємстві. Тематичні, політичні та процедурні теми життєвого циклу розглядаються з метою забезпечення безпеки IoT для підприємства, яка постійно оновлюється та пристосовується до унікальних операційних потреб системи.

2.1 Впровадження та інтеграція

Організації кінцевих споживачів матимуть багато варіантів розгортання функціональних можливостей систем Інтернету речей. Деякі організації самі розробляють системи IoT, проте, існує багато варіантів закупівлі попередньо укомплектованих систем, які включають пристрої із заздалегідь встановленим

програмним забезпеченням та підключенням до мережі, мережні інтерфейси, системи обробки резервної аналітики або деякі їх комбінації.

Системи для безпілотників які розробляються для збору багатьох типів даних з кінцевих точок, щоб передавати ці дані за попередньо налаштованими каналами до шлюзових систем. Шлюзами потім буде передано дані до систем резервної або наземної станції, які надають автоматизовані планування маршруту та потенційну координацію дій для певних типів завдань. Хоча такі системи в ідеалі повинні бути попередньо налаштовані з належною жорсткістю інженерної безпеки під час проектування та розробки, потрібно все-таки виконувати цілий ряд заходів, щоб надійно захищати особливості та вразливості в існуючому підприємстві.

Першим кроком життєвого циклу безпеки є створення концепції безпеки операцій, що відображає дану систему, її потреби в безпеці та методи реалізації в даному напрямку.

2.1.1 Документація про безпеку IoT

Документ про безпеку надає організаціям методичний інструмент, де докладно описані операції з безпеки системи IoT. Документ повинен бути написаний та підтримуватися операторами системи. Приклади шаблонних документів безпеки можна знайти у ряді організацій. Один із прикладів - NIST SP-800-64 [10].

2.1.2 Інтеграція безпеки з мережею

Важко охарактеризувати типову реалізацію мережі IoT, враховуючи, що вона поєднує в собі потенційно так багато різноманітних пристроїв з великою варіацією функцій. Наприклад візьмемо короткий огляд планування щодо інтеграції в мережу та безпеку безпроводового сенсора мереж (WSN – Wireless sensor network) та підключених автомобілів (рис. 2.1).

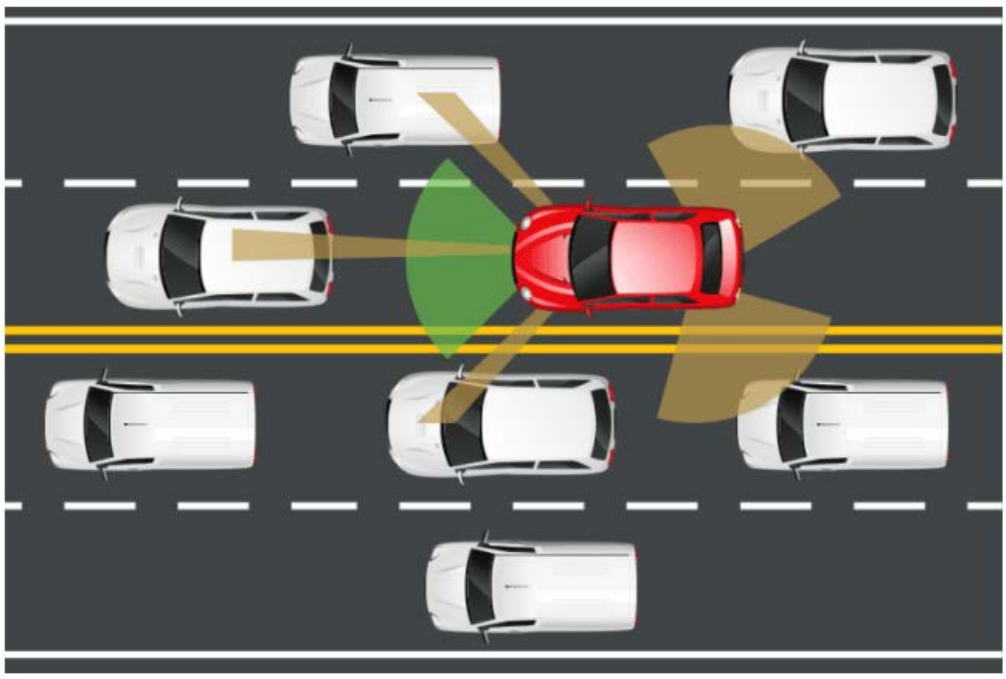

Рисунок 2.1 – Використання сенсорної мережі для забезпечення безпеки руху автомобілів в потоці

2.1.3 Аналіз інтеграції мережі та безпеки для WSN

Вивчаючи типову WSN мережу, ви знайдете багато тисяч або більше малопотужних, датчиків, що працюють на батареях, які ймовірно, спілкуються, використовуючи протокол ZigBee. Ці пристрої можуть спілкуватися на прикладному рівні за допомогою індивідуальних протоколів IoT, таких як MQTT-SN, які можна запускати безпосередньо через ZigBee та подібні стандарти (виключаючи потребу в IP-комутації). У даному сценарії, реалізація MQTT-SN в межах кожного датчика має обов’язково мати шлюз, який прокладається між MQTT-SN (MQTT for Sensor Networks) та протоколом MQTT (Message Queue Telemetry Transport). Шлюз виступає проміжним протоколом між мережею пристроїв IoT та аналітичними системами, що збирають дані з них. Враховуючи, що шлюзи агрегують дані з декількох пристроїв (і часто зберігають дані, принаймні тимчасово), важливо переконатися, що кожен з них має захищену конфігурацію зв’язку, як для кінцевих пристроїв IoT, так і для резервного сервера мережних послуг.

Оглядаючи служби безпеки, необхідні для захисту цих комунікацій, зазвичай використовуються можливості безпеки RF модуля між датчиками і шлюзом. Можна також скористатися можливостями такого протоколу, як TLS (Transport Layer Security — захист на транспортному рівні) між шлюзом MQTT і бекенд-сервісами.

Організаціям не завжди потрібно впроваджувати індивідуальний протокол MQTT-SN, проте деякі пристрої IoT можуть підтримувати можливість безпосередньо спілкуватися за допомогою MQTT зі шлюзом. Наприклад веб-служба Amazon, рішення використовує хмарний шлюз MQTT, що підтримує на пряму такі з'єднання – канали якого захищені за допомогою TLS.

2.1.4 Аналіз інтеграції мережі та безпеки на прикладі підключених автомобілів

Деякі реалізації мають суттєво різні характеристики. Уявіть собі кластер підключених транспортних засобів, з якими кожен спілкується за допомогою протоколу DSRC (Dedicated short-range communications). Ці транспортні засоби відправляють повідомлення один одному та придорожному обладнанню (RSE) багато разів в секунду і залежать від близькості до іншого учасника обміну данними. Ці повідомлення захищені за допомогою можливостей протоколу DSRC, який включає автентифікацію при проходженні даних.

Організаціям часто буде потрібно конфігурувати такі інфраструктурні компоненти і надійно спілкуватися з підключеними транспортними засобами у своїй групі, використовуючи ці протоколи. Незалежно від типу розгортання, ці системи потрібно налаштувати спілкуватися з існуючою технологічною інфраструктурою організації. В перспективі життєвого циклу безпеки, інженери повинні витратити чималий час на планування та інтеграцію. Неправильне планування та впровадження системи IoT в межах підприємства може утворити нові вразливі місця для зловмисників.

2.2 Оновлення мережних та інфраструктурних елементів систем

Цей життєвий цикл включає інтеграційне планування, необхідне для включення нових сервісів та пристроїв IoT в існуючі інфраструктури, що іноді може призвести до значних затрат в плані оновлення застарілої архітектури. Деяким системам IoT для реалізації потрібен зворотній зв'язок в реальному часі для підтримки автоматизованих рішень, хоча початкові втілення IoT будуть зосереджені лише на збиранні даних за допомогою датчиків. Забезпечення аналітики, систем управління та інших функціональних можливостей у межах компанії сприятиме гнучким та масштабованим реалізаціям.

У ситуаціях, коли системи IoT повинні обробляти та діяти відповідно до даних отриманих в реальному часі, необхідно проаналізувати рух в напрямку централізованої обробки даних [11].

Компанія Cisco Systems ввела термін Fog Computing, щоб вирішити необхідність переходу на більш децентралізовану модель, орієнтовану на підвищення надійності, масштабованості та відмовостійкості систем IoT. Модель Fog виконує обчислення, збереження та прикладні послуги на межі мережі або в шлюзах, які обслуговують пристрої IoT [12].

Ця концепція обчислень в гранчних вузлах дозволяє підтримувати початкову аналітику в реальному часі, поліпшення продуктивності та підтримку постійної потреби обміну з централізованими системами. Дані можуть бути локально опрацьовані та проаналізовані з меншою потребою неефективно направляти великі обсяги інформації до додатків в ядрі мережі (рис. 2.2). Після обробки результати отриманих даних можуть бути надіслані безпосередньо до хмари для тривалого збереження або потрапляє в інші програми для аналітики.

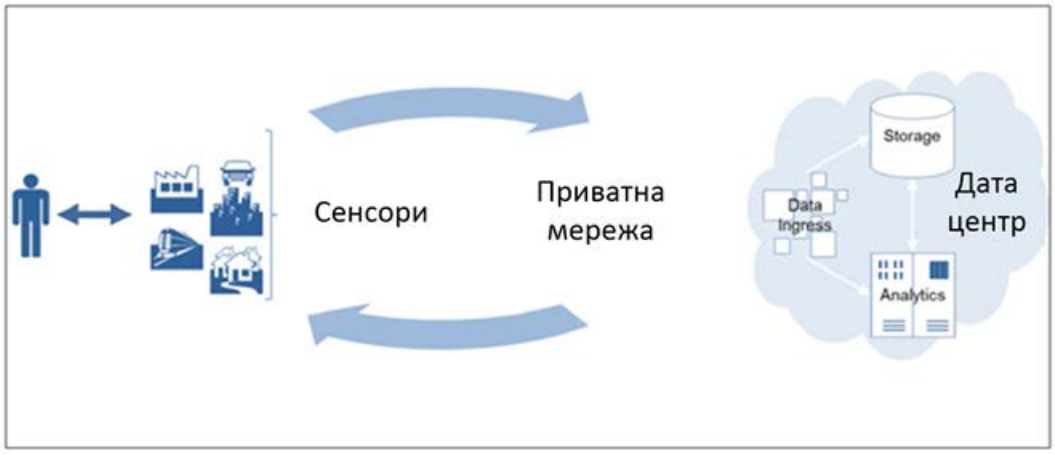

Рисунок 2.2 – Схема реалізації системи шлюзів та гранчних обчислень

Проектування та розгортання систем Інтернету речей, які зможуть масштабуватися і в той же час бути захищеними від нападів, таких як відмова в обслуговуванні (DoS – Denial of Service) є вкрай важливим аспектом. Переосмислення мережної інфраструктури та архітектурної аналітики є важливим кроком на даному етапі.

Децентралізація послуг IoT під час планування та вдосконалення існуючої інфраструктури – це можливість як додати нові послуги, так і підвищити стійкість системи в цілому.

2.2.1 Аналіз механізмів забезпечення безпеки

Інженери також повинні планувати можливість надання мережної інформації, необхідної для IoT пристроїв та шлюзів в нормальному режимі роботи. У деяких випадках це включає планування для розподілу IP-адрес. Вибір підтримуваних протоколів для IoT буде часто диктувати вимоги до IP-адрес. WSN, які використовують такі протоколи зв'язку як Bluetooth, ZigBee і ZWave не вимагають надання IP-адреси, однак протоколи, такі як 6LoWPAN, потребують надання адреси IPv6 для кожного пристрою. Деякі пристрої одночасно підтримують різні безпроводові протоколи та підключення до IP мереж.

Організації, які вирішили надати пристрої з адресами IPv6, стикаються з додатковими завданнями інженерії безпеки, оскільки вони повинні забезпечити маршрутизацію інфраструктури IPv6 надійно і безвідмовно. А також повинні планувати інтеграцію базової необхідної системи доменних імен (DNS). Це потрібно для будь-якої кінцевої точки або шлюзу, який потребує зв'язку за допомогою URL-адрес. Розглянемо протоколи, такі як DNS на основі DNS Суб'єктів (DANE – DNS-based Authentication of Named Entities) для шлюзу інфраструктурного зв’язку. DANE дозволяє значно жорсткіше асоціювати сертифікати з посиланнями (URL – Uniform Resource Locator), використовуючи DNSSEC і може суттєво допомогти стримувати та обмежувати різні сценарії атак MITM на веб-основі.

2.2.2 Інтеграція з системами безпеки

Системи IoT також повинні бути інтегровані з існуючими системами безпеки підприємства, що вимагає взаємодії та тестування інтерфейсів до цих систем. В ідеалі, інтерфейси до цих систем були б створені під час розробки IoT системи, але в деяких випадках спосіб взаємодії має бути розроблений вже в процесі застосування системи.

В інших випадках для взаємодії або використання потрібні прості конфігурації продуктів безпеки цих систем підприємства. Приклади систем безпеки підприємства, з якими, швидше за все, інтегруватимуть IoT, включають наступне:

– системні каталоги;

– системи ідентичності та управління доступом (IAM);

– системи безпеки та управління подіями (SIEM);

– системи управління активами та конфігурацією;

– межі оборонних систем (наприклад, брандмауери, системи виявлення);

– криптографічні системи управління ключами;

– системи контролю бездпроводового доступу;

– існуючі аналітичні системи.

2.3 Інфраструктура передавання даних в системах IoT

Крім систем IOT на основі безпроводових мереж, існують також системи на базі IoT з включенням на шини даних для зв'язку із сусідніми пристроями. Наприклад, у сучасних автомобілях зазвичай шина мережі контролера (CAN) використовується для обміну повідомленнями між компонентами транспортного засобу в реальному часі (рис. 2.3). В недалекому минулому виробники автомобілів почали впроваджувати розширену функціональність на основі мультимедійних платформ транспортних засобів. У багатьох випадках це реалізовано зв'язками між цими новими системами (наприклад, інформаційно-розважальними системами) та критичними для безпеки шинами CAN. Високий рівень безпеки вимагає, що такі системи мають бути відокремленими, однак, навіть коли відбувається сегрегація, критично вразливі місця CAN шини можуть залишитися відкритими для атаки.

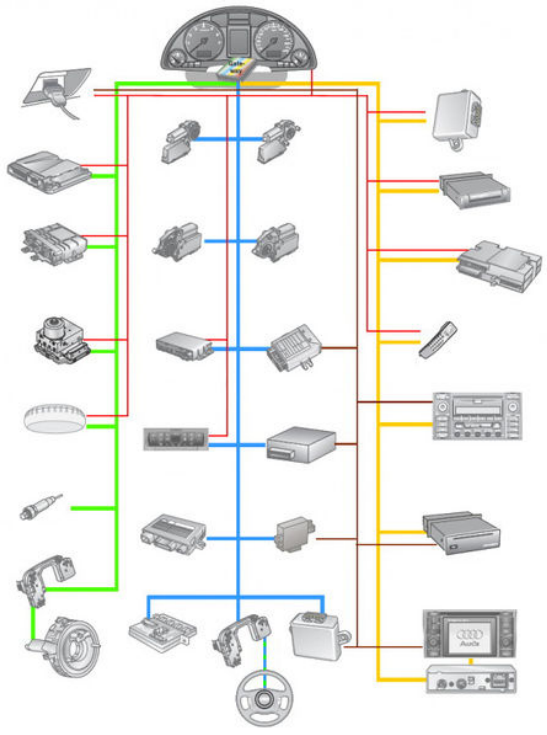

Рисунок 2.3 – Структура обміну інформацією в реальному часі через шину мережі автомобіля Audi

Вивчаючи дослідження, проведені Чарлі Міллером та Крісом Валасеком у 2015 році, можна зрозуміти деякі проблеми, з якими стикаються транспортні засоби. Через неправильну конфігурацію несучої мережі, погану архітектуру безпеки в межах програмного забезпечення та інженерної основи одного з MCU (відповідає за сегментацію інформаційно-розважальної системи транспортного засобу від критичних для безпеки шин CAN), дослідники змогли ефективно дистанційно керувати підключеним транспортним засобом [17].

У ситуаціях, коли системи IoT інтегруються в критичні для безпеки системи, розділення на окремі одиниці є життєво важливим. Це означає, що застосовуються методи сегментації чутливих функцій від нечутливих. Крім того, надання підтримки захисту цілісності, автентифікації, захисту від повторних повідомлень, та конфіденційніості доцільна у багатьох випадках. У традиційних мережах SIEM інтеграція таких модулів має вирішальне значення для перевірки трафіку та забезпечення дотримання правил щодо даних, які перетинають встановлені зони безпеки. Аналогічні системи потрібні в майбутньому, які працюватимуть в реальному часі через шини даних.

2.4 Контроль безпеки системи

Для перевірки потрібно провести достатнє тестування, як позитивне, так і негативне, щоб функціональні вимоги безпеки були задоволені. Це тестування повинно виконуватися в операційному середовищі після інтеграції системи з іншими компонентами інфраструктури підприємства. В ідеалі це тестування відбувається протягом всього життєвого циклу розвитку, а також інтеграції, розгортання та використання.

Перевірка дає гарантії, що система працює відповідно до набору вимог, які належним чином відповідають потребам зацікавлених сторін. Перевірка – це впевненість у тому, що продукт, послуга або система IoT відповідає потребам клієнта та іншим параметрам зацікавлених сторін – в системі IoT це означає, що при

проектуванні систем має закладатися достатній рівень безпеки. Перевірка – це оцінка того, чи відповідає продукт, послуга чи система регулюванням, вимогам, специфікаціям або обмеженням, які накладаються ринком. Для IoT системи, це означає, що служби безпеки та захисту були реалізовані згідно з дизайном та архітектурою проекту на стадії розробки [13].

Один із підходів до перевірки функціональних вимог безпеки – це створення тестових макетів або емулятори, які користуються функціоналом кінцевих систем, прораховуючи все це в режимі реального часу на віртуальній машині. Наприклад, створення емулятора, який емулює екземпляр захищеного з'єднання (наприклад, TLS) та автентифікацію між пристроями, забезпечить розробникам впевненість, що кожен пристрій працює відповідно до визначених вимог безпеки в різних ситуаціях та сценаріях використання.

Тестування системи необхідне для перевірки відповідності функціональним вимогам безпеки конкретного впровадження IoT, що має бути досягнуто під час розробки та інтеграції. Тестування IoT системи повинно бути максимально автоматизованим і має стосуватися різних варіантів поведінки системи.

Про виявлені розбіжності та помилки мають створюватися звіти, та відстежуватися до закриття групами розробників по мірі оновлення системи або доступності нових релізів. Відстеження помилок та збоїв можна проводити у середовищі різних інструментів відстеження від формальних інструментів управління конфігурацією, таких як DOORS до більш серйозних інструментів, таких як Jira.

2.5 Навчання з безпеки

Open DNS за 2018 рік у звіті Enterprise дав ранній погляд на виклики, з якими незабаром зіткнуться фахівці з безпеки. У звіті це було визначено співробітниками які вже постачають власні пристрої IoT на підприємство і з'ясували, що пристрої, такі як смарт-телевізори, простягаються через все підприємство і є шлюзами для різних інтернет-сервісів. Це дослідження висвітлює аспект потреби перекваліфікувати

працівників та адміністраторів безпеки в тому, що доцільно приєдувати до мережі, а також як ідентифікувати належним чином приєднані пристрої IoT.

Створення тренінгу з безпеки вимагає періодичного огляду та можливого створення нових політик безпеки, необхідних для підтримки різних парадигм IoT. Ці політики повинні використовуватися як вихідний матеріал для навчання з підвищення обізнаності, а також навчання адміністрації з питань безпеки [14].

Системи IoT часто мають унікальні характеристики, які не зустрічаються в традиційних IT системах. Теми для розгляду у навчанні з питань безпеки користувачів мають включати наступне:

–дані, мережні та фізичні ризики, пов'язані з пристроями IoT;

–політика щодо залучення до організації персональних пристроїв IoT;

–вимоги щодо захисту конфіденційності даних, зібраних на пристроях IoT;

–порядок взаємодії (якщо можливо) з корпоративними пристроями IoT.

Адміністратори з питань безпеки повинні бути забезпечені технічними та процедурними джерелами інформації, яка необхідна для безпечної роботи систем IoT.

2.6 Безпечні конфігурації

Системи інтернету речей містять багато різноманітних компонентів і кожен повинен бути налаштований безпечним способом. Кожен компонент також повинен бути налаштований на взаємодію з іншими компоненти через захищені з’єднання. Часто легко не помітити необхідність зміни налаштувань за замовчуванням і обрати правильні режими безпеки для роботи. Завжди необхідно аналізувати існуючі конфігурації безпеки, щоб зрозуміти, як заблокувати систему IoT та послуги зв'язку в разі необхідності.

2.6.1 Конфігурації пристроїв IoT

Деякі з більш потужних пристроїв IoT використовують операційну систему в режимі реального часу (RTOS – Real-time operating system), що вимагає перегляду файлів конфігурації та налаштувань за замовчуванням. Деякі функції, наприклад, завантаження операційної системи, повинні бути переглянуті та оновлені таким чином, що дозволяє лише автентифіковані та захищені цілісністю оновлення програмного забезпечення. Необхідно переглянути відкриті порти, протоколи та заблокувати всі, які не потрібні в процесі експлуатації. Крім того, слід використовувати налаштування портів за замовчуванням, коли можливо реалізувати елементи управління білими програмами. Коротше кажучи, необхідно створити базовий рівень захисту за замовчуванням для кожного типу пристрою.

Не менш важливою є безпека конфігурацій обладнання. Як було обговорено вище, необхідно блокувати будь-які відкриті тестові інтерфейси (наприклад, JTAG) для боротьби з можливістю зловмисника отримати доступ до пристроїв, які викрадені або піддаються впливу. Також використовують будь-які функції фізичної безпеки, які можуть бути включеними в апаратне забезпечення. Такі функції можуть включати активне виявлення несанкціонованих дій та відповідей (наприклад, автоматизоване витирання конфіденційних даних при підробці), охоплення і блокування критичних інтерфейсів та ін.

Також важливою є безпечна конфігурація протоколу. Потрібно аналізувати літературу, що стосується протоколів для IoT або стеків протоколів, та дотримуватися основних вимог при розгортанні систем IoT. З прикладами інструкцій щодо безпечної конфігурації Bluetooth IoT можна ознайомитися в наступних випадках:

– управління інформаційним забезпеченням Агенції національної безпеки (IAD), посібник з безпеки Bluetooth [16];

– посібник з безпеки Bluetooth по NIST SP 800-121 [17].

Часто узгодження щодо використання елементів безпеки не береться до уваги виробниками, в результаті чого компоненти IoT постачаються з незахищеними

конфігураціями за замовчуванням. Наприклад, для протоколу ZigBee використовуються профілі додатків, які підтримують сумісність між різними реалізаціями ZigBee. Ці профілі додатків містять ключі за замовчуванням які необхідно змінити перед запуском в роботу системи. Тобіас Зілнер та Себастьян Стробл надали корисний інструктаж щодо необхідності змін параметрів за замовчуванням. Дослідники відзначили, що за замовчуванням параметри Trust Center Link для ZigBee Light Link Profile (ZLL), та для ZigBee Home Automation Public додатків (HAPAP) засновані на парольній фразі ZigBeeAlliance09. Реалізація будь-якої системи IoT, яка не застосовує модифікацію ключів за замовчуванням, може зробити багато інших засобів безпеки непотрібними в межах підприємства. Ці ключі слід завжди оновлювати до того, як онлайн-мережа IoT на базі ZigBee буде введена в експлуатацію [15].

2.6.2 Безпечна конфігурація шлюзу та мережі

Після оновлення конфігурації захисту на пристроях IoT необхідно проаналізувати конфігурацію пристроїв які виступають шлюзами, та взаємодіють з кінцевими точками IoT. Шлюзи – це агрегатори численних пристроїв IoT, тому особливу увагу слід приділити їх безпеці та конфігурації. У деяких випадках ці шлюзи розміщуються на локальних носіях та пристроях IoT, але в інших випадках пристрої IoT можуть безпосередньо спілкуватися з шлюзами, розташованими у хмарі (наприклад AWS IoT).

Одним з важливих аспектів конфігурації шлюзу є те, як вони реалізують безпеку та комунікацію з нижніми та верхніми рівнями. Шлюз являє собою зв’язок із інфраструктурою ядра мережі та завжди повинен бути налаштований для запуску TLS або іншого VPN-з'єднання (наприклад, IPSec), в ідеалі потрібна двостороння (взаємна) автентифікація на основі сертифікатів. Це вимагає, щоб передача даних з інфраструктури, з якою взаємодіє шлюз, була налаштована з належним доступом та елементами керування, які базуються на сертифікаті шлюзу. Часто недооціненим

аспектом цих конфігурацій є міцність допустимих підтримуваних шифрів. Далі рекомендується використовувати останні версії TLS. Наприклад, на момент написання, TLS 1.2 слід використовувати замість TLS 1.1 або 1.0, оскільки обидві попередні версії мають опубліковані вразливості. TLS 1.3 наразі перебуває у статусі проекту IETF. Як тільки його буде доопрацьовано та реалізація стане широко доступною, їх слід прийняти в експлуатацію.

Окрім шифрів, шлюзи спілкуються з іншими серверами додатків, тому слід переконатися, що послуга пов'язана із сертифікатом PKI. Один спосіб досягнення цього, згаданого раніше, полягало у використанні DANE, протоколу, в якому DNSSEC використовується разом із записами DANE для перевірки кореляції цифрових даних сертифікатів на сервері. DANE був створений для зменшення кількості реальних PKI загроз при розгортанні, пов'язаних з неправдивими сертифікатами спільно з DNS.

Висновки до розділу

У цьому розділі проаналізовано комплексний підхід до безпеки систем Інтернету речей, який тісно пов'язаний з процесами розробки, інтеграції та розгортання. Життєвий цикл безпеки систем Інтернету речей повинен підтримувати структурні рішення, такі як:

– політика конфіденційності як потенціал витоку конфіденційної інформації або метаданих через сторонні застосунки, що вимагає всебічного контролю конфіденційності;

– велика кількість нових пристроїв та типів пристроїв, які повинні бути надійно налаштовані для захисту від нових векторів атак на підприємство;

– автономні операції та транзакції з пристрою на пристрій, які зменшують вплив атак;

– ризики, пов'язані з безпекою, яким IT-персонал традиційно не піддавався. Ці ризики можуть призвести до шкоди для співробітників та клієнтів, якщо зловмисник компроментує систему IoT з можливістю заподіяти фізичну шкоду;

– попередня обробка та початкова аналітика даних.

У кожному з них є життєво важливі підпроцеси, які повинні бути узгоджені або запроваджені в інтеграцію систем, які містять елементи IoT в різних галузях.

3 ВИКОРИСТАННЯ МЕРЕЖНИХ СХОВИЩ ДЛЯ ПОКРАЩЕННЯ БЕЗПЕКИ В СИСТЕМАХ ІНТЕРНЕТУ РЕЧЕЙ

3.1 Мережні сервіси та IoT

Що стосується розгортання B2B, споживчих та промислових систем Інтернету речей, ніщо не з'єднує пристрої, дані пристроїв, людей та організації разом більше, ніж хмарні рішення для IoT та допоміжні послуги. Шлюзи, програми, брокери протоколів та різноманітні компоненти даних для аналітики та бізнес-аналітики для зручності перебувають у хмарі, що має низку переваг які заключаються у вартості та масштабованості таких систем. З точки зору підтримки мільярдів пристроїв IoT хмарні послуги пропонують найбільш сприятливе середовище для нових або вже існуючих компаній. У відповідь CSP (CSP – Cloud Solution Provider) почали пропонувати все більше і більше функцій підтримки безпечного з'єднання продуктів IoT. З’являється все більше стандартизованих комплексів, щоб допомогти компаніям, що пропонують товари та послуги IoT, почати розгортання на базі хмарних технологій з мінімальними зусиллями. Організації, які йдуть шляхом стандартизації цих хмарних рішень, усвідомлюють важливість впровадження засобів безпеки, які мають бути вбудовані в кожну пропозицію.

Наприклад, ARM нещодавно працював з Freescale та IBM над створенням хмарного IoT-стартового комплекту [15]. У комплект входить MCU, який автоматично передає дані на веб-сайт в інтернеті. Хоча набір орієнтований на підготовку розробників для легкого впровадження хмари в рішення IoT, важливо, щоб розробники розуміли, що процес виробництва дуже різниться і вимагає індивідуального інженерного процесу безпеки.

У цьому розділі розглядаються деякі мережні сервіси, які починають виступати на підтримку впровадження систем IoT. З таким підходом планується найближчим часом розгорнути мільйони продуктів IoT в різних системах, хмара є оптимальним механізмом відстеження місцезнаходження та стану цих пристроїв. Дані хмарні рішення надають підтримку в забезпеченні безпеки пристрою, оновленні програмного

забезпечення та контролю актуальної конфігурації. Враховуючи можливість безпосередньо впливати на функціональний стан та безпеку IoT пристрою, безпека в даних послугах є найвищим пріоритетом.

Одним з найважливіших аспектів безпеки IoT є можливість відстежувати активи та товарні запаси. Сюди входять також атрибути пристроїв. Хмара – велике рішення, що включає управління активами та запасами підприємства, що забезпечує перегляд на всіх пристроях, які були зареєстровані та уповноважені працювати в межах організації.

3.1.1 Надання послуг, виставлення рахунків та права управління

Надання послуг, виставлення рахунків та права управління ─ цікавий випадок використання, оскільки багато постачальників пристроїв IoT пропонують свої пристрої клієнтам як послугу. Для цього потрібна можливість відстежувати підписки, авторизувати або видалити авторизацію для роботи пристроїв, а також підготувати рахунки у відповідь до кількості використання послуги яка надається додатково до пристроїв IoT. Приклади включають послуги передплати для камери та інших систем моніторингу на основі сенсорів (наприклад, запис даних відеоспостереження в хмарне сховище DropCam), моніторинг та відстеження (наприклад, послуги пристроїв FitBit) та багато інших.

3.1.2 Моніторинг у реальному часі

Мережні програми, що використовуються для підтримки критично важливих можливостей, таких як управління у надзвичайних ситуаціях, промисловий контроль та виробництво можуть забезпечити моніторинг у реальному часі. Де це можливо, багато організацій починають забезпечувати моніторинг та інші функції в хмарі щоб зменшити експлуатаційні витрати, зробити більш доступними дані та відкривають нові B2B та B2C послуги. Оскільки кількість кінцевих точок IoT збільшується, ми

побачимо пристрої, наприклад, програмовані логічні контролери (PLC) та віддалені термінальні блоки (RTU) які встановлюватимуть прямий зв’язк із хмарою, підтримуючи можливість контролювати системи все більш ефективно.

3.1.3 Координація датчиків

Взаємозв’язок від машини до машини пропонує розширені можливості координувати та рівномірно автономно проводити обслуговування. З часом все більше робочих процесів ставатимуть автоматизованими, все більше виключаючи людей з життєвого циклу пристроїв. Хмара буде відігравати центральну роль у реалізації цих автоматизованих робочих процесів. Як приклад – в хмарі з’являться служби, яку пристрої IoT зможуть отримувати по запиту, щоб зібрати останню інформацію, обмеження чи інструкції. Налаштування доступу до протоколів, які керують багатьма IoT системами (наприклад, MQTT), забезпечує можливість створення все більшої кількості сценаріїв реалізації.

3.1.4 Обмін інформацією

Однією з головних переваг IoT є те, що він дозволяє обмінюватися інформацією для багатьох зацікавлених сторін. Наприклад, імплантований медичний пристрій може надавати інформацію до медичного кабінету і цей медичний кабінет може потім надати дану інформацію страховому постачальнику. Інформація також може зберігатися в поєднанні з іншою інформацією, зібраною про пацієнта. Обмін інформацією та послуги інтероперабельності хмари є обов'язковими передумови для забезпечення потужної аналітики IoT. Враховуючи різноманітність апаратних засобів IoT платформи, сервісів та структур даних, такі провайдери, як wot.io, мають на меті забезпечити послуги обміну даними на рівні програмного забезпечення для безлічі джерел і постачальників даних. Багато додатків IoT та допоміжні протоколи публікуються і поширюються, відповідаючи загальнопоширеним форматам

програмного забезпечення, які можуть інтерпритуватися в різні мови програмування. Такі послуги мають вирішальне значення для включення даних B2B, B2I та пропозиції B2C.

3.1.5 Транспорт та трансляція повідомлень

Хмара та її централізовані, пристосовані, еластичні можливості – ідеальне середовище для реалізації широкомасштабних послуг транспорту повідомлень IoT. Багато хмарних сервісів підтримують протоколи HTTP, MQTT та інші протоколи, які в різних комбінаціях, можуть транспортувати, транслювати, публікувати дані, або переміщувати дані іншими необхідними способами (центрально або на межі мережі). Однією із основних перешкод в обробці даних IoT є управління масштабом. IoT потребує архітектурної здатності хмари для еластичного масштабування своїх даних та послуг – отже, послуги транспорту та трансляції повідомлень є необхідними для задоволення безпрецедентних і зростаючих вимог.

3.2 Дослідження пропозицій мережних послуг IoT

Пропозиції безпеки на основі хмар, які також називаються безпекою як послуга (SECaaS – Secuity as a service), представляють швидко зростаючий в об’ємах хмарний бізнес і ці пропозиції дійшли до рівня для підтримки IoT. Пропозиції SECaaS не тільки масштабуються, але й допомагають організаціям справлятися з проблемами, викликаними обмеженим технологічним рівнем безпеки, який закладений на фізичному рівні у пристроях. У більшості компаній сьогодні бракує людей та знань, необхідних для виконання інтеграції безпеки в бізнес, щоб бути в курсі останніх загроз безпеці, та здійснювати моніторинг безпеки доцільно використовувати рішення які пропонують на сьогодні CSP.

3.2.1 AWS IoT

Amazon готовий стати провідним розповсюджувачем хмарних IoT-сервісів та в багатьох випадках буде постачальником хмарних послуг IoT. Власні слова компанії Amazon:

«AWS IoT – це керована хмарна платформа, яка дозволяє легко підключати пристрої, надійно взаємодіяти з хмарними програмами та іншими пристроями. AWS IoT може підтримувати мільярди пристроїв і трильйони повідомлень і вони можуть обробляти та направляти їх повідомлення до кінцевих точок AWS та інших пристроїв надійно та безперебійно» [2].

AWS IoT Amazon є основою, яка дозволяє пристроям IoT спілкуватися з хмарою, використовуючи різні протоколи (HTTP, MQTT тощо). В даній концепції, пристрої IoT можуть спілкуватися між собою та сервісами через програми-брокери. AWS IoT інтегрується з багатьма іншими службами Amazon. Наприклад, можна використовувати свій механізм потокового передавання даних та аналітики в реальному часі, Kinesis. Kinesis Firehose функціонує як платформа прийому, приймаючи потоки даних і завантажуючи їх в інші домени Amazon, служба простого зберігання (S3 – Simple Storage Service), Redshift (зберігання даних) та еластичний пошук Amazon (ES – Elastic Search). Потрапивши на відповідну платформу даних, різноманітну аналітику можна провести за допомогою Kinesis Streams. Amazon Glacer [5] забезпечує масштабоване, довгострокове архівування даних та резервне копіювання для важливої інформації.

Що стосується підтримки програм та розробки, AWS IoT (рис. 3.1) інтегрується та добре поєднуються з Amazon Lambda, Kinesis, S3, CloudWatch, DynamoDB та різними інші хмарними службами Amazon.

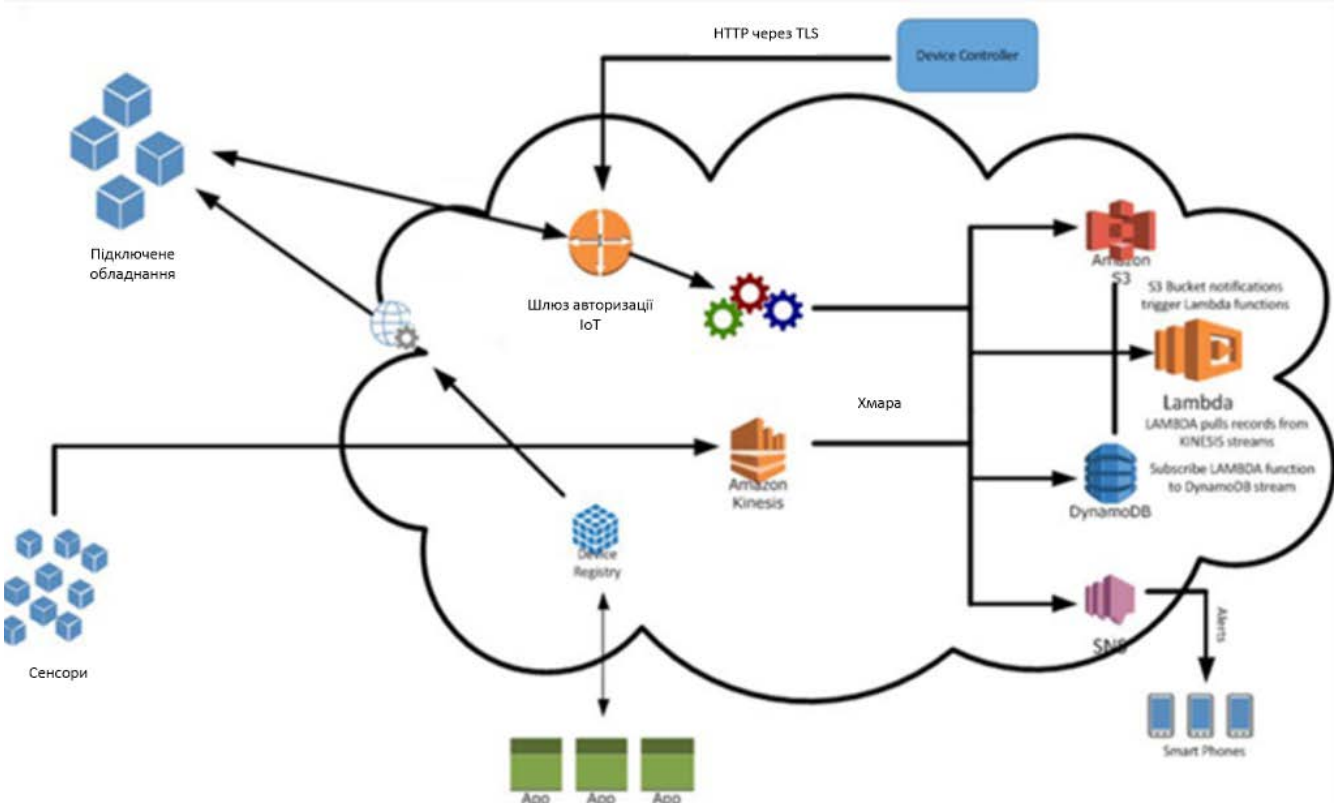

Рисунок 3.1 – Інтеграція AWS IoT з іншими сервісами Amazon

Платформа Amazon IoT почала залучати різні галузі, в тому числі і охорону здоров'я. Наприклад, Philips співпрацює використовуючи послуги AWS IoT як двигун своєї платформи Health Suite Digital. Ця платформа розроблена так, щоб вона дозволяла постачальникам медичних послуг та пацієнтам взаємодіяти новими революційними та гнучкими способами, використовуючи прилади охорони здоров'я IoT, традиційні джерела даних, аналітику та звітність. Багато інших компаній, пов’язаних з IOT, починають користуватися AWS або співпрацювати з ними.

Послуги CSP, такі як AWS IoT, пропонують можливість попередньо налаштувати пристрої Інтернету речей та потім завантажити конфігурації на фізичні пристрої, коли вони готові вийти в онлайн. Після запуску AWS IoT пропонує віртуальну систему Thing Shadow, яка може підтримувати стан вашого пристрою IoT навіть у режимі офлайн. Стан конфігурації зберігається в документ JSON, що зберігається в хмарі. Так, наприклад, якщо лампочка з включеним MQTT офлайн, команда MQTT (рис. 3.2) може бути відправлена до сховища віртуальних речей, щоб

змінити її колір. Коли лампочка повернеться в Інтернет, вона змінить свій колір відповідним чином, виконавши команду, яка була в черзі очікування.

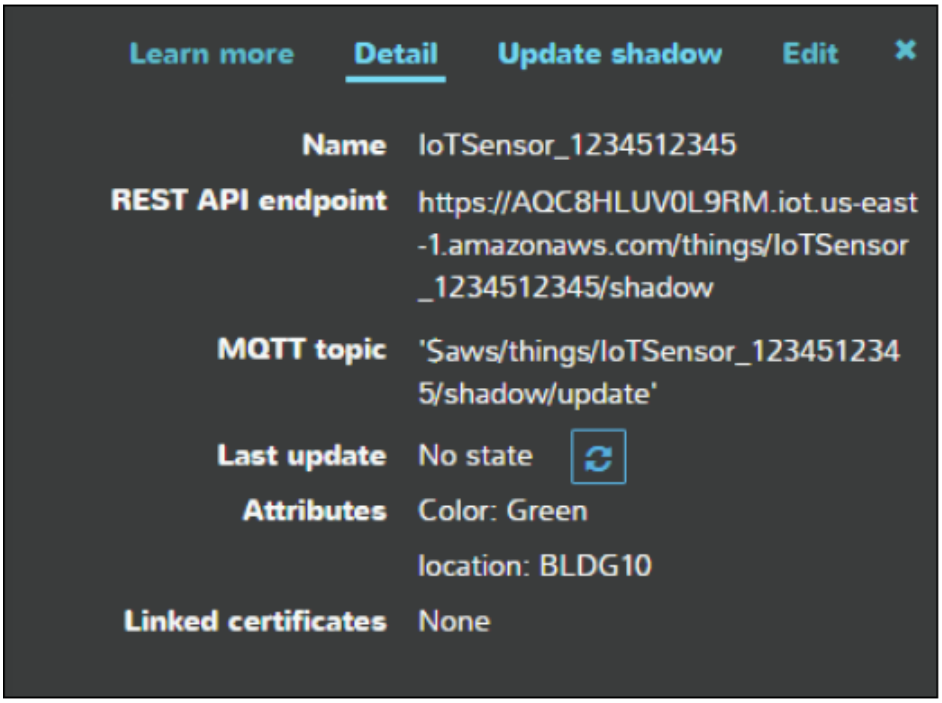

Рисунок 3.2 – Налаштування для віддаленого управління лампочкою через AWS IoT

AWS Thing Shadow є посередником між контрольною програмою та пристроями IoT. Дані програми використовують протокол MQTT з попередньо визначеними темами які можна використовувати для взаємодії зі службою та пристроями. MQTT-повідомлення, які є зарезервовані для послуги Thing Shadow починаються з $ aws/things/thingName/shadow.

Нижче наведені зарезервовані команди MQTT, які можна використовувати для взаємодії з Shadow Thing [6].

– /update;

– /update/accepted;

– /update/documents;

– /update/rejected;

– /update/delta;

– /get;

– /get/accepted;

– /get/rejected;

– /delete;

– /delete/accepted;

– /delete/rejected.

AWS IoT публікує JSON документ для кожного оновлення та відповідає на кожне оновлення або отримання запиту статусом /прийнято або /відхилено.

З точки зору безпеки, важливо, щоб тільки авторизовані кінцеві точки та програми могли встановлювати зв’язок. Також важливо, щоб адміністративна консоль була достатньо захищена від несанкціонованого доступу, коли зловмисники отримують доступ до безпосередньо налаштування та управління IoT пристроями. Щоб проілюструвати деякі робочі процеси з обробки даних AWS IoT (рис. 3.3), необхідно оглянути додатковий випадок використання для підключеної системи, яка використовує обробку даних та можливості хмари AWS.

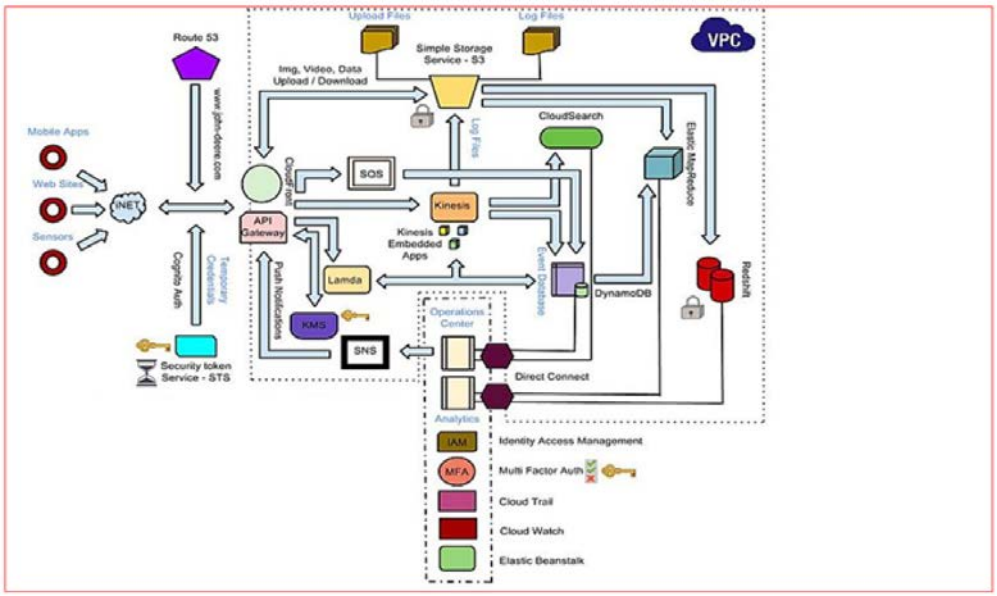

Рисунок 3.3 – Схематичне зображення процесу обробки даних в системі AWS IoT

У цьому випадку використання існує низка кінцевих точок, які вводять дані в AWS хмару. Дані надходять до AWS через ряд потенційних вхідних шлюзів:

– Kinesis;

– Kinesis Firehose;

– MQTT broker.

Потрапивши в середину AWS, платформа визначає згідно правил та політик, куди слід перенаправляти дані і які додаткові дії потрібно вжити до них. У багатьох випадках дані будуть надсилатися до бази даних – наприклад, S3. Redshift також може бути використаний і його слід використовувати для збереження записів з часом, а також для тривалого зберігання даних.

У наборі AWS IoT можна скористатися інтегрованим управлінням журналом функцій через CloudWatch. CloudWatch можна налаштувати безпосередньо в межах AWS IoT для реєстрації подій процесу у повідомленнях, що надходять від пристроїв до AWS інфраструктури. Журнал повідомлень може бути встановлений на помилки, попередження, інформаційні або відлагоджувальні повідомлення (рис. 3.4).

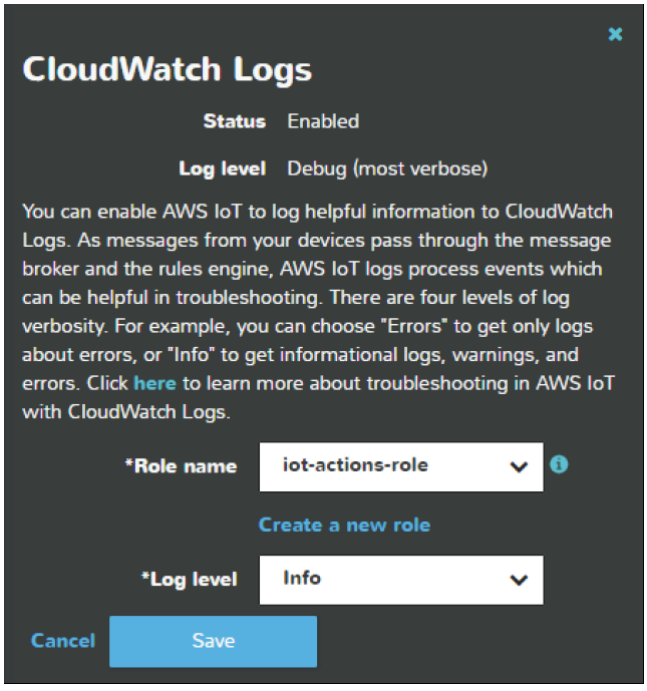

Рисунок 3.4 – Вікно конфігурації повідомлення відлагодження у CloudWatch

Amazon CloudTrail також слід використовувати для розгортання IoT на базі AWS. CloudTrail підтримує API AWS на рівні облікового запису, щоб увімкнути аналіз безпеки, аналітику та відстеження відповідності. Є багато сторонніх журналів

управління таких систем, як Splunk, AlertLogic та SumoLogic, які інтегруються безпосередньо в CloudTrail.

3.2.2 Microsoft Azure IoT

Microsoft також зробила великий стрибок в розвитку в бік хмарних сервісів IoT зі своїм Azure IoT Hub. Azure може похвалитися деякими потужними функціями управління пристроями IoT для інтеграторів та розробників таких систем, включаючи оновлення та налаштування програмного забезпечення, політик безпеки та використання кінцевих пристроїв. Поза межами управління пристроєм IoT, Azure надає функції, що дозволяють розробникам IoT організовувати та групувати пристрої в межах своїх операційних доменів (рис. 3.5). Іншими словами, це дозволяє управляти топологією на рівні пристрою, а також конфігурувати на кожному пристрої, необхідні параметри, встановити управління на рівні групи, дозволи та контроль доступу. Служба управління групою Azure надається через API групи пристроїв, в той час як функції управління пристроєм, версією програмного забезпечення та забезпечення – надаються через API управління його реєстром пристроїв.

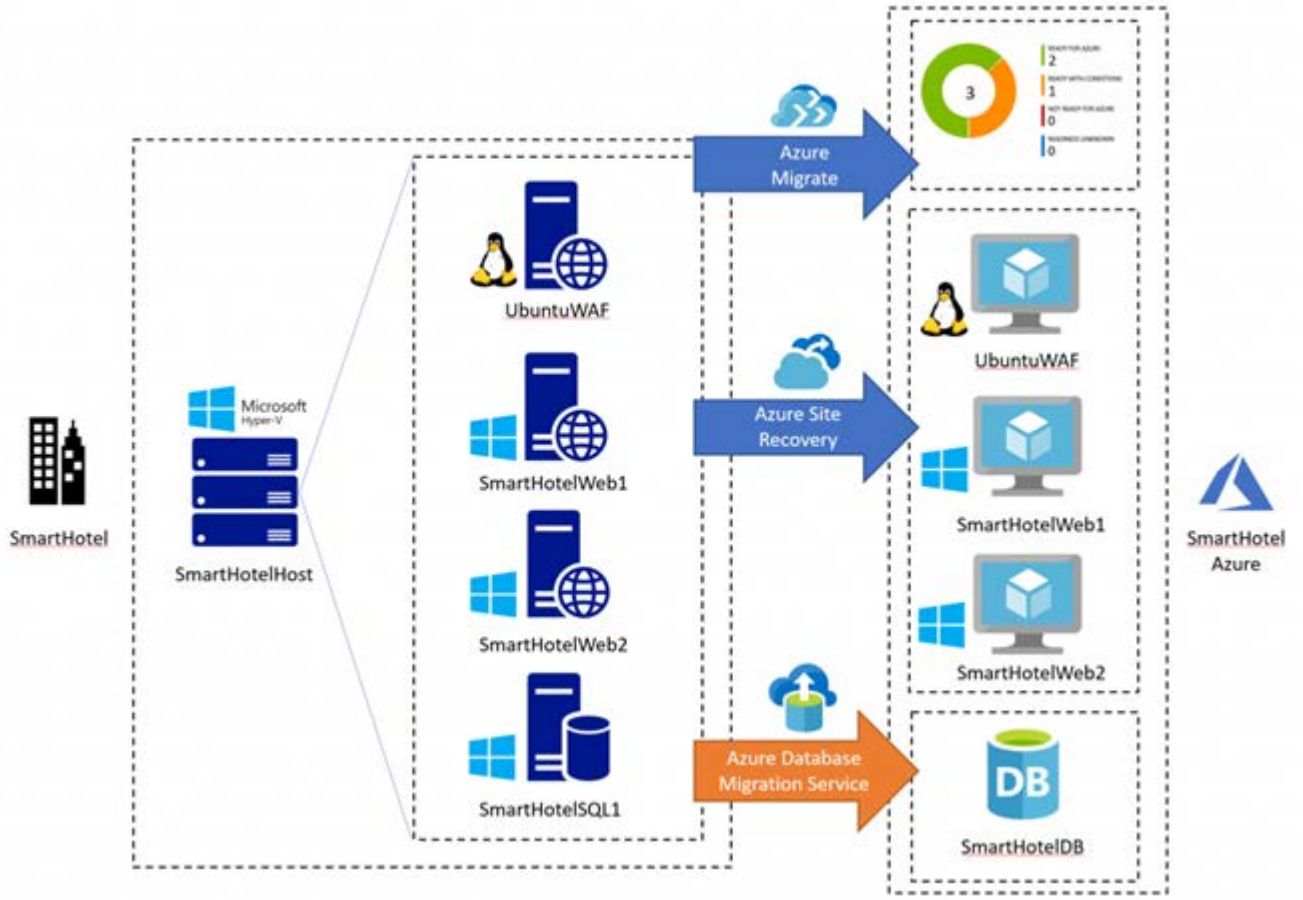

Рисунок 3.5 – Приклад схеми взаємодії через Microsoft Azure

Централізована автентифікація надається за допомогою існуючого каталогу Azure Active. Azure IoT Hub підтримує протоколи, пов'язані з IoT, такі як MQTT, HTTP та AMQP, щоб налаштувати зв’язок хмара-пристрій. З огляду на різноманітність стандартів зв'язку, Azure забезпечує крос-протокольну взаємодію Fusion для розробників через загальний формат повідомлень IoT Hub. Формат повідомлення складається з різноманітних полів властивостей системи та додатків. Якщо потрібно, щоб комунікації між пристроєм та хмарою могли використовувати існуючий центр подій Azure API, але якщо потрібна автентифікація на кожному пристрої та контроль доступу, IoT Hub забезпечить цю можливість.

Автентифікація на пристрої та контроль доступу в Azure увімкнено через використання маркерів безпеки IoT Hub, які відображають політику доступу та облікові дані кожного пристрою. Автентифікація на основі токена дозволяє проводити

автентифікацію без передавання чутливих параметрів безпеки по всьому з’єднанню. Токени засновані на унікальному ключі, створеному Azure, який генерується за допомогою виробника, або ідентифікатора пристрою, наданим виробником. Щоб проілюструвати деякі робочі процеси обробки даних Azure IoT, звернемося до конфігурації резервного копіювання в Azure. Як і у випадку з AWS, для підключених пристроїв існують різні точки входу в хмару. Дані можна вводити в Azure через шлюз API або через IoT сервісні платформи, які підтримують REST та MQTT. Дані потім можуть бути надіслані до блоку зберігання або до файлу. Також мережа доставки вмісту Azure (CDN) – це надійний та зручний інструмент для розповсюдження оновлень мікропрограмного забезпечення у інвентарі пристроїв IoT.

3.2.3 Cisco Fog Computing

IoT-стратегія Cisco для хмарних технологій вирішує той факт, що переважна більшість IoT пристроїв працюють на межі мережі, які близькі до ядра мережі. Зі свого боку Cisco, намагається надати набагато потужніші функціональні ресурси та безпеку, закладені в граничних елементах мережі, які мають велику кількість підключених джерел даних. Переваги зберігання даних та обробки якомога центральнішими платформами можна описати наступними твердженнями:

– скорочена затримка: багато інтенсивних даних опрацьовуються в режимі реального часу оскільки вони включають величезну кількість сенсорних даних;

– ефективність передавання даних в мережі: обсяги даних, що містять IoT сервіси, є вкрай великими і є багато випадків, коли перенесення даних не має сенсу;

– політика може бути локально керована;

Попередні переваги, мабуть, найбільш відчутні для промислового IoT, де не обійтися лише централізованою обробкою даних. Часові потоки сенсорів, контролери та виконавчі пристрої, програми моніторингу та звітності, об'ємні набори даних, пов'язані з промисловим IoT, роблять Fog модель обчислення досить привабливою.

Туманні обчислення Cisco, хоча і на початку свого життєвого циклу, вже впроваджені в проміжне програмне забезпечення, яке знаходиться між апаратним забезпеченням та запущеними програмами безпосередньо на кінцевому обладнанні.

Основна архітектура IOx складається з наступного: вузли туману, вони представляють пристрої (наприклад, маршрутизатори та комутатори), які містять крайові мережі та надають хост-ресурси в рамках системи. Хост-операційна система, яка надає наступні можливості:

– Cisco Application Framework (CAF) для місцевого застосування;

– управління та контроль;

– додатки (багатьох можливих типів);

– мережні та проміжні послуги.

Адміністратор мережі підключений до API CAF, забезпечує централізоване управління додатками та сховищем і працює на всіх вузлах системи.

Розробка IoT Fog Computing підтримується програмним забезпеченням Cisco DevNet. Набори для розробки IoT систем також можуть включати використання існуючих рішень Cisco в галузі кібербезпеки, такі як Cisco NetFlow, TrustSec та інші.

3.2.4 IBM Watson IoT platform

IBM Watson та його можливості були представлені світу ще в 2010 році, коли платформи когнітивних обчислень Watson почали перемагати найкращих чемпіонів на знаменитому ігровому шоу Jeopardy. Пізнавально навчальна обчислювальна здатність вчитися і вирішувати проблеми використовуються з користю у різних галузях, таких як охорона здоров'я.

Сьогодні IBM розширює домен обробки Watson, застосовуючи його до Інтернету речей. Основоположні інтерфейси API IoT від IBM доступні через IBM центр розвитку платформи Watson IoT: «https://developer.ibm.com/ iotfoundation/» та «https://developer.ibm.com/iotfoundation/recipes/ api-documentation/» і включають можливості взаємодії в Інтернеті речей, такі як:

– інвентаризація та перегляд IoT-пристроїв організації;

– реєстрація, оновлення та перегляд пристроїв;

– операції на поглинутих наборах даних.

Транспорт та комунікація пристроїв IoT сприяють підтримці платформи протоколів зв'язку MQTT та REST [3] що дозволяє розробникам IoT створювати потужні системи прийому та обробки даних, виконувати когнітивну аналітику та реалізовувати можливості виведення даних.

API MQTT платформи Watson IoT дозволяє не шифрувати з'єднання на порту 1883. Та зашифровані комунікації на портах 8883 або 443. Слід зазначити, що платформа вимагає TLS 1.2. Рекомендовані IBM шифри такі як:

– ECDHE-RSA-AES256-GCM-SHA384;

– AES256-GCM-SHA384;

– ECDHE-RSA-AES128-GCM-SHA256;

– AES128-GCM-SHA256.

Реєстрація пристроїв вимагає використання підключення TLS коли по MQTT пароль передається назад клієнту, використовуючи захищений тунель TLS.

Коли MQTT використовується для підключення пристрою до хмари, існує можливість використовувати маркер замість пароля MQTT. У цьому випадку значення use-token-auth присвоюється замість пароля.

Інтерфейс REST також захищений TLS 1.2. Допустимий порт – 443 і ключ програми API служить іменем користувача, тоді як маркер автентифікації використовується як пароль на підтримку базової автентифікації HTTP.

3.3 Хмарний контроль безпеки систем IoT

Враховуючи різноманітність хмарних сервісів, які підтримують розгортання IoT, кожна хмара і кінцева точка користувача або пристрою відіграє важливу роль у забезпеченні безлічі політик та правил.

Нижче наведено короткий перелік рекомендованих засобів безпеки IoT та послуги, які організація повинна врахувати при розгортанні систем. Основні елементи управління, такі як автентифікація та шифрування до хмари підтримуються усіма CSP. Більшість CSP постачають послуги різними способами. Використання тої чи іншої організації може бути безпосередньо чи опосередковано визначеним для отримання та використання послуг на основі унікальних пакетів та пропозицій. Ці послуги можуть поєднуватися різними способами для створення потужних, захищених з'єднань у вашій віртуалізованій інфраструктурі.

3.3.1 Автентифікація та авторизація

Беручи до уваги контроль безпеки аутентифікації при інтеграції систем Інтернету речей для забезпечення безпеки необхідно буде дотримуватися наступних пунктів:

1. Перевірити осіб, які мають доступ до адміністративних функції та API (тут надається перевага багатофакторній аутентифікації).

2. Автентифікувати кінцевих користувачів на хмарні програми.

3. Автентифікувати хмарні програми (включаючи шлюзи та брокери IoT) від одного до іншого.

4. Безпосередньо автентифікувати пристрої IoT (які мають необхідну безпеку та функціональні ресурси) для шлюзів та брокерів.

5. Проксі-автентифікація кінцевих користувачів від постачальника програм до іншого.

CSP підтримують різноманітні механізми аутентифікації. Amazon AWS і Microsoft Azure описані в наступних розділах.

3.3.2 Amazon AWS IAM

Служба аутентифікації AWS IAM, що підтримується хмарою Amazon, є багатофункціональною платформою аутентифікації, яка підтримує об'єднану ідентичність, багатофакторну автентифікацію, управління користувачем / роллю / дозволом та включає повну інтеграцію з іншими послугами та сервісами Amazon.

Служба багатофакторної аутентифікації (наприклад, на основі токена) AWS (MFA) служби IAM підтримує різноманітні чинники форм зовнішньоекономічної діяльності, що відповідають будь-яким стандартам вашої організації та створюють нову або використовують існуючу базу аутентифікації. Жетони обладнання, брелоки, картки доступу і віртуалізовані пристрої MFA (наприклад, ті, які можуть працювати на мобільному пристрої) підтримуються Amazon. MFA може використовуватися як ваша віртуальна приватна хмара адміністраторів, а також кінцевих користувачів.

Перехідна авторизація користувачів протікає між кількома веб-додатками, що реалізовано за допомогою відкритого стандарту OAuth2.0 (RFC6749) для авторизації, що дозволяє забезпечити безпечний, делегований доступ до сторонніх веб-сервісів. Однак OAuth2 забезпечує лише доступ до авторизації. Функціональність аутентифікації можна отримати за допомогою побудованої послуги OpenID Connect (OIDC) на OAuth2 OIDC використовує ідентифікаційні маркери, через OAuth2 транзакції з підтримкою авторизації для користувачів.

3.3.3 Автентифікація Azure

Як було сказано раніше, Microsoft Azure надає централізовану та об'єднану ідентифікацію також через Azure Active Directory (AD).

Microsoft Azure також пропонує послуги OAuth2 та OpenID Connect як ідентифікатори послуги, в рамках своєї пропозиції Azure AD. Amazon AWS пропонує таку можливість, як і частину ідентифікації та пропозиції управління доступом. Якщо ваш обраний постачальник хмарних послуг не пропонує OpenID Connect, але

пропонує OAuth2, ви також можете інтегрувати послугу OAuth2 від іншого постачальника, хоча це може бути не настільки зручним рішенням як при використанні єдиного сервіс провайдера.

3.3.4 Оновлення програмного забезпечення / мікропрограмного забезпечення

Величезна кількість вразливих версій у стеках виконання програмного та мікропрограмного забезпечення можна нейтралізувати за допомогою швидких, легких та високо автоматизованих за стосунків для виправлення. Рекомендується впроваджувати автоматизовану, захищену систему оновлення програмного забезпечення для кінцевих користувачів. Нові оновлені файли або виконавчі фрагменти (патчі) повинні мати цифрові підписи у вашому середовищі DevOps за допомогою затвердженого програмного забезпечення. Що стосується кінцевих пристроїв, ви повинні переконатися, що програмне забезпечення та оновлення мікропрограмного забезпечення, що поширюються на кінцеві пристрої IoT, може перевірити ці кінцеві пристрої. Деякі CSP підтримують послуги оновлення програмного забезпечення / мікропрограмного забезпечення, такі як Azure CDN тощо.

Щоб мати впевненість у облікових даних, якими користується певний пристрій потрібно аутентифікувати послуги та шлюзи, та слід бути обережними під час надання довіри та прав пристроям. Залежно від критичності конкретного пристрою, програма для завантаження може бути отримана від постачальника або особисто від надійного агента. Завершення завантаження та ініціалізація надає оперативні сертифікати пристроїв у захищеному порядку (і по мережі).

3.3.5 Моніторинг стану безпеки системи

Шлюзи / брокери IoT повинні бути налаштовані на пошук підозрілої поведінки кінцевих точок та їх безперервний моніторинг. Наприклад, брокери MQTT повинні фіксувати повідомлення від серверів і клієнтів, які можуть сигналізувати про

зловмисну поведінку. У версії специфікації MQTT 3.1.1 наведено приклади такої поведінки для повідомлення:

– неодноразові спроби з'єднання;

– неодноразові спроби аутентифікації;

– ненормальне припинення з'єднання;

– тематичне сканування;

– надсилання недоставних повідомлень;

– клієнти, які підключаються, але не надсилають дані.

3.4 Побудова захищеної архітектиру IoT з використанням хмарних сховищ

Існує багато архітектурних аспектів і варіантів створення хмарної IoT-системи. CSP постачальники послуг IoT та підприємства, повинні вивчити можливості які надаються для архітектурної побудови відповідних засобів безпеки в конкретних умовах. На наступній схемі представлена загальна віртуальна приватна система з хмарним сервісом постачальника, який пропонує основні функціональні та охоронні послуги для захисту кінцевих та проміжних вузлів даних. Він показує типові, віртуалізовані послуги, доступні для загальних IT систем, а також розширення на базі IoT. Не всі розгортачі IoT повинні використовувати всі доступні хмарні можливості, але для більшості потрібен мінімальний набір перерахованих вище послуг та вимоги до їх надійного захисту (рис. 3.6).

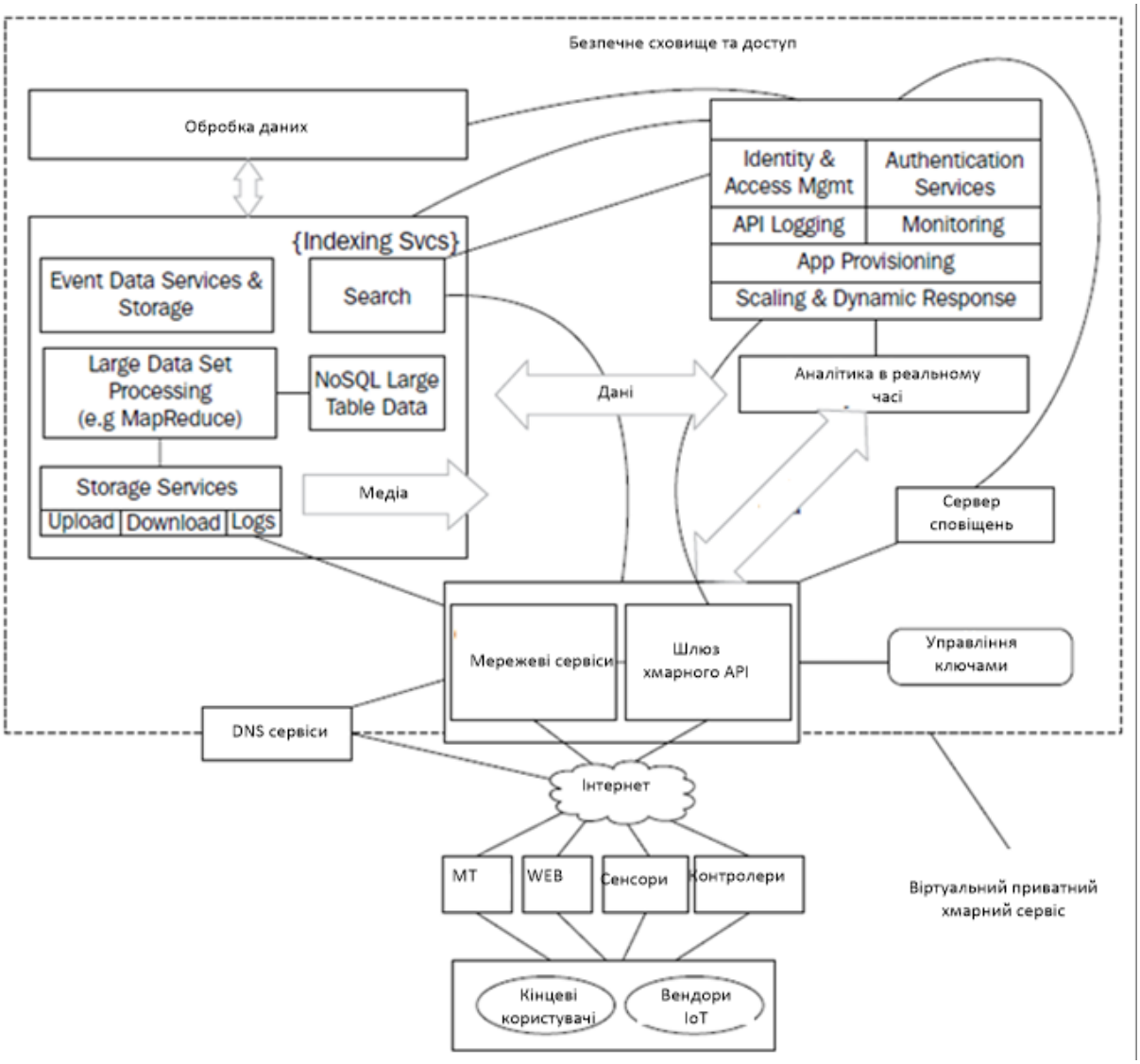

Рисунок 3.6 – Загальна віртуальна приватна система з хмарним сервісом

Варто також перелічити позитивні характеристики хмарних рішень для IoT систем, а також перспективи на майбутнє, з використанням підключенння до хмар.

Хмара має багато описаних вище характеристик, які роблять її привабливою, адаптивною та реалізує стек технологій, за допомогою якого можна передбачити, побудувати та розгорнути нові сервіси IoT.

3.4.1 Програмне забезпечення, визначене мережею (SDN)

SDN з'явилися як можливості управління мережею нового покоління для спрощення та зменшити обсягу роботи над переналаштуванням мереж та управлінням маршрутами на основі визначених політик. Іншими словами, вони були створені для того, щоб зробити саму мережу більш програмованою
і динамічною, що абсолютно необхідно для величезних масштабів і гнучкості, для керування світовим трафіком IoT. SDN архітектури функціонують за допомогою роз'єднання мережі контролю від функції маршрутизації. Вони складаються з контролерів SDN, які включають:

– API або міст, який з'єднується з мережними програмами;

– API Southbound, який підключає мережні контролери до мережних пристроїв, які виконують маршрутизацію трафіку.

IoT архітектури, які використовують великі хмарні сервіси, вже користуються SDN. Великі системи віртуалізації, на яких розміщуються сервери управління, брокери, шлюзи, польові пристрої IoT та інші елементи архітектури IoT вбудовані в Amazon, Google та інші хмарні провайдери. З очікується, що ми побачимо набагато більш гнучкі можливості у здатності створювати, адаптувати та динамічно налаштовувати мережі IoT. SDN використовуються сьогодні постачальниками безпеки для вирішення проблем з розподіленою відмовою у наданні послуг (DDOS), на які слід звернути увагу підприємствам та адаптувати їх реалізацію для підтримки цієї функціональності.

3.4.2 Послуги передавання даних

З огляду на велику кількість даних, джерел даних та потоків даних в IoT, хмарне середовище забезпечує інструменти для управління та структурування цих даних. Наприклад, DynamoDB Amazon пропонує надзвичайно масштабований, низький рівень затримки, NoSQL можливості бази даних для операцій з даними та зберігання,

обмін та аналітику IoT послуг. За допомогою зручного веб-інтернету розробники створюють та керують таблицями, журналами, доступом та іншими функціями контролю даних. Користь для організацій IoT будь-якого розміру полягає в тому, що моделі ціноутворення пропорційні кількості фактично використоаних даних.

Захист даних, автентифікація та контроль доступу можуть бути реалізовані на кожному вузлі в DynamoDB, використовуючи систему ідентифікації AWS та систему управління доступом. Це означає, що одна організація може виконувати різноманітну аналітику, виробляти похідні дані, заповнені у різних таблицях, а потім вибірково робити ці дані доступними через додаток для багатьох унікальних клієнтів.

Один з викликів, з яким стикаються в середовищах розробки IoT, це різноманітність апаратних платформ IoT. Різноманітні платформи поставляються з різним програмним забезпеченням, API та драйверами. Мови програмування, які використовуються в усіх регіонах, на обладнанні також варіюється, від C до Python та багатьох інших.

Середовище розробки для багаторазового використання, яке може бути спільним, є викликом для команди розробників, які мають реалізувати досить гнучке ПЗ для використання в різноманітних сценаріях конфігурування систем.

Один із підходів до підтримки дуже гнучкого середовища розробки IoT за допомогою використання контейнерної технології. Використовуючи цю технологію, контейнери можуть бути побудовані з бібліотеками та пакетами, необхідними для розробки поточного типу пристрою. Ці контейнери можна копіювати та ділитись всією командою розробників як базовим інструментом. Оскільки команда розробляє нові типи пристроїв IoT, для використання можуть бути створені нові базові лінії, які додають нові стеки та бібліотеки програмного забезпечення.

3.5 Перехід до підключення 5G

У той час як США, Європа та Азія узгоджують свої відмінності у формулюванні як належить визначити стандарт 5G, обіцяють ряд його важливих особливостей які революційно збільшать кількість пристроїв Інтернету речей. Покриття мережі через 5G стане ключовим фактором розвитку Інтернету речей у своїй здатності підтримувати на порядок більше пристроїв при значно більшій швидкості передавання даних, ніж мережі LTE. Поки що існують такі погляди на специфікацію 5G [4]:

– швидкість передавання даних повинна починатися від 1 Гб/с;

– затримка повинна бути меншою за 1 мс;

– обладнання 5G повинно бути набагато енергоефективнішим, ніж його попередники;

Враховуючи простір IP-адрес IPv6 та найближче майбутнє 5G підключення, не дивно, що багато компаній, що заздалегідь мислять, вкладають гроші і наполегливо готуються до немислимого зростання для IOT.

Висновки до розділу

У цьому розділі представлено мережні сервіси та архітектурні рішення безпеки, розроблені для підтримки Інтернету речей. Використовуючи хмарні сервіси та дотримуючись основних аспектів безпеки, організації можуть керувати та розгортати багатодоменні системи IoT. Забезпечення різних аспектів та параметрів IoT також вимагає визначення, які елементи безпеки є обов'язками для замовника перед мережним постачальником, для цього необхідно проаналізувати основні параметри:

– контроль безпеки Cloud IoT: які елементи необхідні для того щоб створити ефективну архітектуру безпеки IoT для підприємства;

– налаштування архітектури безпеки IoT: аналіз наявних пропозиції хмарної безпеки для поєднання та суміщення в ефективну, загальну архітектуру безпеки;

– обчислювальні технологіях IoT: аналіз обчислювальних потужностей які надають мережні платформи.

Досліджено мережні пропозиції та пропозиції щодо безпеки веб-служб Amazon Web Services (AWS), компоненти, пропоновані Cisco (Fog Computing), а також Microsoft Azure, які можуть бути ефективно інтегровані в системи Інтернету речей, обираючи вендора в залежності від потреб замовника.

4 ПРАКТИЧНА РЕАЛІЗАЦІЯ ПІДКЛЮЧЕННЯ ПРИСТРОЮ ІНТЕРНЕТУ РЕЧЕЙ ЧЕРЕЗ СЕРВІС AWS IOT

AWS IoT встановлює високу планку безпеки. Сьогодні для цього потрібна безпека транспортного рівня (TLS) версії 1.2 зі взаємною автентифікацією. Раніше вимоги до обчислень означали, що певні пристрої не мали достатньої кількості оперативної пам’яті чи процесора для прямого підключення. Ці пристрої зазвичай підключаються до AWS IoT через пристрій шлюзу або проксі-сервер, який буде обробляти безпеку та автентифікацію від їхнього імені. ESP8266 був обмежений в цьому плані.

Перший крок нашого шляху в Інтернеті – перейти від моделі "безпека через проксі" та перенести взаємну аутентифікацію та шифрування в наш мікроконтролер. У цьому розділі йде мова про те, як взяти власний ESP8266 (або ESP32) та підключити його безпосередньо до AWS IoT за допомогою Mongoose OS. Mongoose OS – це операційна система з відкритим кодом для мікроконтролерів, яка реалізує підключення до хмари. Його розробили Cesanta, заснована в Дубліні компанія з вбудованим програмним забезпеченням та Advanced APN Technology Partner.

4.1 Поетапна реалізація підключення пристрою IoT до мережного сервісу

Покроково в цьому розділі, ми отримаємо: сертифікати на наш ESP8266, які будуть зареєстровані на AWS IoT та зберігатимуться на вашому локальному комп’ютері. Зображення вбудованого ПЗ для ESP8266, яке може підключитися безпосередньо до AWS IoT і може бути налаштоване за допомогою кінцевої точки AWS IoT, пароля Wi-Fi SSID, не змінюючи його середовище побудови, яке можна використовувати для розробки нових мікропрограм.

Базовий набір елементів необхідних для реалізації даного рішення:

– AWS CLI налаштований з дійсним обліковим записом AWS, який ви маєте намір використовувати;

– плата NodeMCU на базі ESP8266 (ви також можете використовувати ESP32);

– USB-кабель для підключення плати NodeMCU до комп'ютера;

– mos – інструмент Mongoose OS;

– клон git сховища Mongoose OS від GitHub, який ви можете отримати за допомогою git-клона командою: https://github.com/cesanta/mongoose-os.

Всі команди в цих інструкціях повинні виконуватися в одній оболонці командного рядка. Припускається, що додаток «mos» вже інстальований перед початком налаштувань. Якщо це не так, потрібно буде посилатися на нього безпосередньо або додавати його до свого каталогу.

Додаток «mos» створить політику AWS IoT у вашому обліковому записі під назвою «mos-default». Це відкрита політика лише для цілей розвитку. Не рекомендовано використовувати цю політику у виробництві чи в обліковому записі, який обробляє дані про виробництво. Якщо ви хочете створити власну політику, див. Приклад політики в документації AWS IoT, щоб дізнатися більше.

Якщо програма «mos» не виявить ваш послідовний порт, прочитайте розділ «Виправлення неполадок - встановлення послідовних драйверів» після підручника.

Крок 1. Створення файлу операційної системи Mongoose.

1. Перейдемо до сховища git у нашій оболонці.

2. Далі перейдемо до каталогу прикладного програмного забезпечення c_mqtt:

```
$ cd fw/examples/c_mqtt
```

3. Створюємо прошивку:

```
$ mos build --arch esp8266
Connecting to http://mongoose.cloud, user test
Uploading sources (3007 bytes)
Success, built c_mqtt/esp8266 version 1.0 (20170315-154447/???).
Firmware saved to build/fw.zip
```

4. Завантаження прошивки:

```
$ mos flash
```

```
Loaded temp/esp8266 version 1.0 (20170313-194120/???)
Using port /dev/cu.SLAB_USBtoUART
Opening /dev/cu.SLAB_USBtoUART...
Connecting to ESP8266 ROM, attempt 1 of 10...
  Connected
Running flasher @ 460800...
  Flasher is running
Flash size: 4194304, params: 0x0240
Deduping...
    2544 @ 0x0 -> 0
  553248 @ 0x11000 -> 483616
  131072 @ 0xdb000 -> 20480
     128 @ 0x3fc000 -> 0
Writing...
    4096 @ 0x1000
   77824 @ 0x11000
   12288 @ 0x25000
   98304 @ 0x29000
    4096 @ 0x44000
   12288 @ 0x4a000
   61440 @ 0x4e000
   20480 @ 0x5f000
    4096 @ 0x65000
   12288 @ 0x67000
   24576 @ 0x6b000
  159744 @ 0x72000
   20480 @ 0xdb000
    4096 @ 0xfb000
Wrote 516096 bytes in 12.76 seconds (316.02 KBit/sec)
Verifying...
    2544 @ 0x0
```

5. Налаштуємо Wi-Fi, відповідно замінивши параметр YOUR_WIFI_SSID та YOUR_WIFI_PASSWORD на відповідні значення для нашого оточення:

```
$ mos wifi YOUR_WIFI_SSID YOUR_WIFI_PASSWORD
Using port /dev/cu.SLAB_USBtoUART
```

```
Getting configuration...
Setting new configuration...
Saving and rebooting...
```

6. Згенеруємо сертифікати, завантажимо їх на плату NodeMCU та встановимо параметри MQTT, замінивши REGION на ім'я регіону, який ми хочемо використовувати з AWS IoT:

```
$ mos aws-iot-setup --aws-region REGION --aws-iot-policy mos-default
Using port /dev/cu.SLAB_USBtoUART
AWS region: us-east-1
Connecting to the device...
Current MQTT confit: {
  "clean_session": true,
  "keep_alive": 60,
  "pass": "",
  "pub": "/response",
  "reconnect_timeout_max": 60,
  "reconnect_timeout_min": 10,
  "server": "axxxxxxxxxxxxz.iot.us-east-1.amazonaws.com:8883",
  "ssl_ca_cert": "ca-verisign-ecc-g2.crt.pem",
  "ssl_cert": "aws-iot-exxxxxxxxf.crt.pem",
  "ssl_cipher_suites": "",
  "ssl_key": "aws-iot-exxxxxxxxf.key.pem",
  "ssl_psk_identity": "",
  "ssl_psk_key": "",
  "sub": "/request",
  "user": "",
  "will_message": "",
  "will_topic": ""
}
Generating certificate request, CN: mos-M3SRGEJDINMOPB12
Generating private key locally
Asking AWS for a certificate...
Certificate ID: bxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx8
```

```
Certificate ARN: arn:aws:iot:us-east-1:0xxxxxxxxx8:cert/
bxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx8
Wrote private key to aws-iot-bxxxxxxxc.key.pem
Wrote certificate to aws-iot-bxxxxxxxc.crt.pem
Attaching policy "Mongoose" to the certificate...
Uploading certificate...
Uploading key...
Uploading CA certificate...
New config: {
  "clean_session": true,
  "keep_alive": 60,
  "pass": "",
  "pub": "/response",
  "reconnect_timeout_max": 60,
  "reconnect_timeout_min": 10,
  "server": "axxxxxxxxxxxxz.iot.us-east-1.amazonaws.com:8883",
  "ssl_ca_cert": "ca-verisign-ecc-g2.crt.pem",
  "ssl_cert": "aws-iot-bxxxxxxxc.crt.pem",
  "ssl_cipher_suites": "",
  "ssl_key": "aws-iot-bxxxxxxxc.key.pem",
  "ssl_psk_identity": "",
  "ssl_psk_key": "",
  "sub": "/request",
  "user": "",
  "will_message": "",
  "will_topic": ""
}
Setting new configuration...
Saving and rebooting...
```

7. Відкриємо CLI для контролю пристрою (рис. 4.1).

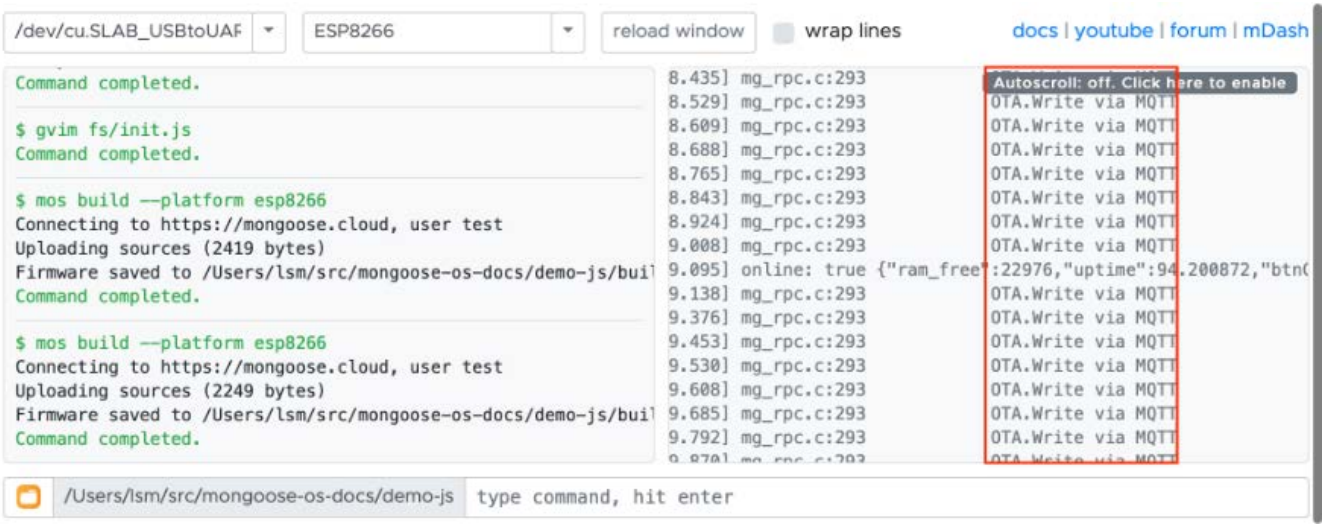

Рисунок 4.1 – Command Line Interface

Крок 2. Взаємодія з пристроєм

1. Входимо до консолі AWS IoT і обираємо «Тест», щоб перейти до клієнта MQTT.

2. У розділі «Опублікувати» встановлюємо тему /request.

3. У текстовій області під темою вводимо:

```
{
    gpio: {
        pin: 2,
        state: 0
    }
}
```

4. Обираємо кнопку «Опублікувати до теми».

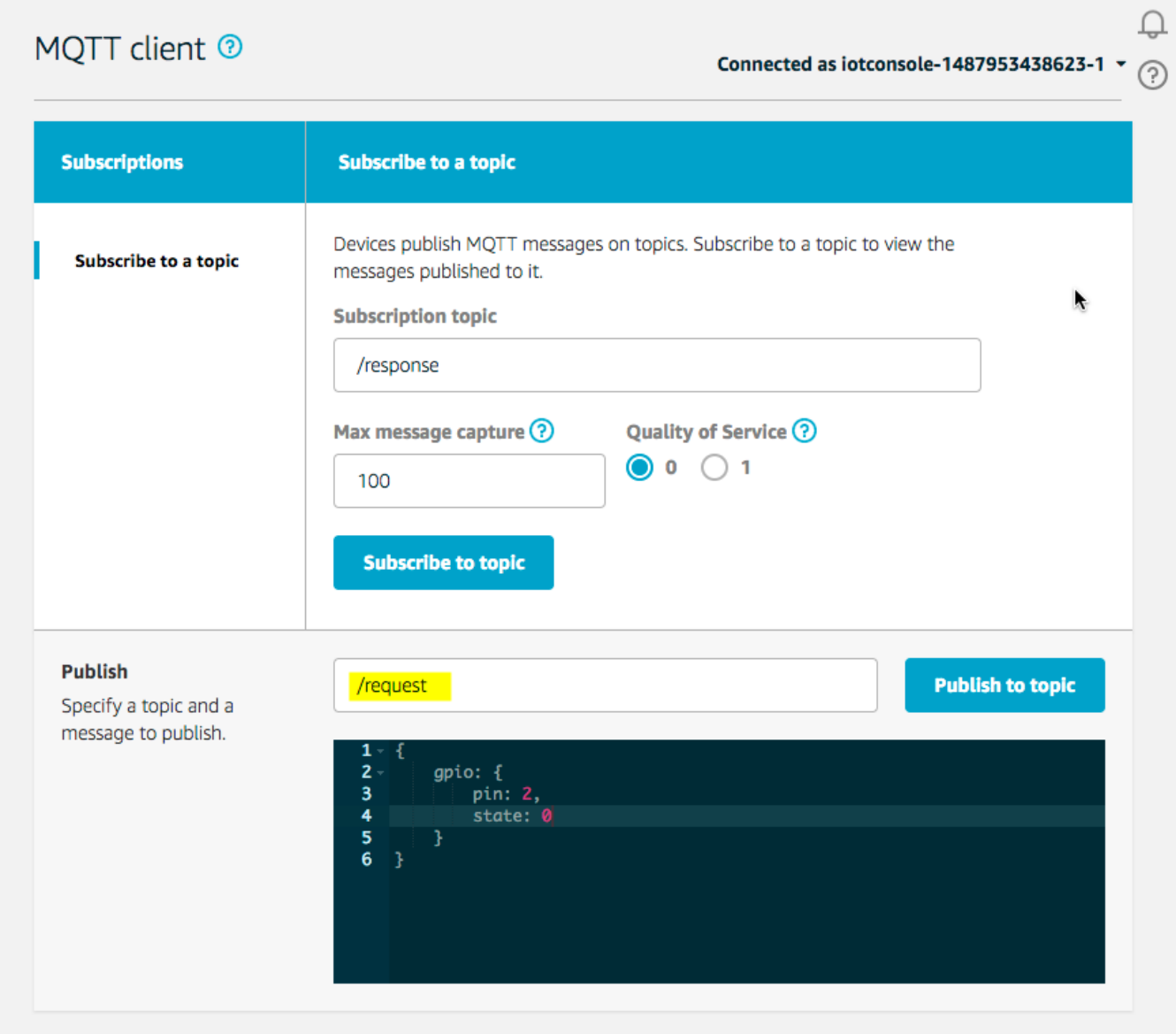

Рисунок 4.2 – Налаштування MQTT клієнта

У цей момент увімкнеться вбудований синій світлодіод на платі NodeMCU.

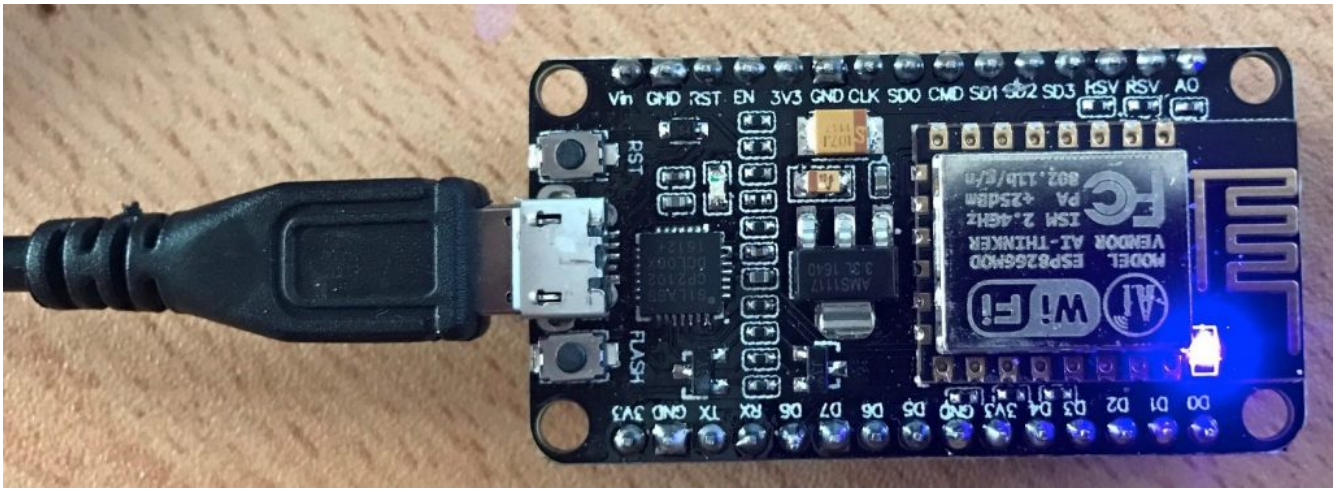

Рисунок 4.3 – Вигляд плати ESP8266

Синій світлодіод увімкнено на GPIO2, який інвертується, тож встановивши його у стан 0, увімкнеться синій світлодіод (рис. 4.3).

5. У текстовій області під типом теми:

```
{
    gpio: {
        pin: 2,
        state: 1
    }
}
```

4.2 Опис основних структурних елементів даного рішення

4.2.1 Підтримка ESP32

Mongoose OS підтримує декілька архітектур. Все, що було описано – поки що буде працювати і на платах ESP32. Єдине, що потрібно буде змінити – каталог архіву та назву серійного пристрою. Для ESP32 будується прошивка за допомогою цієї команди:

```
mos build --arch esp32
```

4.2.2 Виправлення неполадок. Встановлення послідовних драйверів

Якщо послідовний порт не було виявлено автоматично, можливо, доведеться встановити драйвер послідовного порту. Необхідно оглянути плату NodeMCU та визначити її серійний номер (рис. 4.4). Дві найпоширеніші мікросхеми – це CP2102 та CH340.

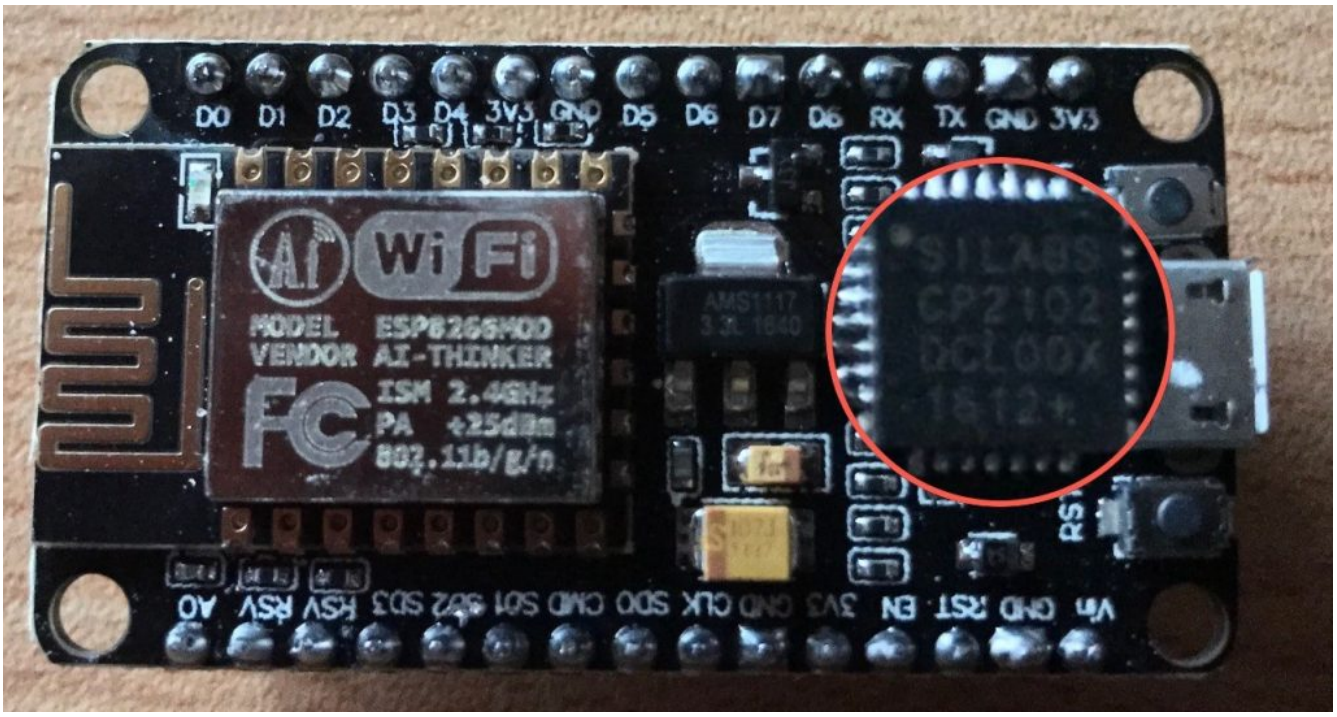

Рисунок 4.4 – Серійний номер чіпу на платі

Якщо це плата CP2102, потрібно завантажити серійні драйвери Silicon Labs.

Якщо, відповідно, плата CH340, завантажується відповідний драйвер для операційної системи. Зараз для цього компонента передбачені драйвери Windows та драйвери MacOS.

4.2.3 Виділена криптографія

Багато мікроконтролерів не мають ні спеціальних криптографічних інструкцій, ні захищених елементів. Відсутність належних криптографічних інструкцій означає, що деякі криптографічні операції мають істотний вплив на енергоспоживання та, як правило, забирають багато часу. Відсутність захищених елементів означає, що сертифікати TLS мікроконтролера можна витягувати з апаратного забезпечення, копіювати та використовувати для представлення себе за пристрій, використовуючи легко доступні інструменти для налагодження.

Пристрій ATECC508A CryptoAuthentication ™ від Microchip Technology поєднує криптографічні функції на основі апаратних засобів та захищене зберігання в конструкції, яка чинить опір атаці через фізичні, електричні та програмні засоби. Пристрій підключається через інтерфейс I2C до мікроконтролера. Потім мікроконтролер використовує простий набір команд для виконання криптографічних операцій над даними приватним ключем, який залишається на ATECC508A. ATECC508A може внутрішньо генерувати приватні ключі або зберігати приватні ключі, створені зовнішньою системою. Під час розробки продукту ця зовнішня система може бути комп'ютером розробника. При повному обсязі виробництва ця зовнішня система, як правило, є високошвидкісним модулем захисту апаратних засобів (HSM), встановленим у захищеному виробництві.

Виключаючи необхідність хост-процесора для обробки криптографічних операцій, ATECC508A може допомогти підвищити безпеку та продуктивність. На основі мікроконтролерів, що використовують ATECC508A, можна встановити TLS-з'єднання швидше, ніж реалізація TLS лише для програмного забезпечення.

AWS тісно співпрацював з Microchip і Cesanta, щоб створити спосіб використання пристрою ATCHC508A Microchip з ESP8266 і ESP32 на платформі Mongoose OS Cesanta. У цьому розділі ми детально проходимо цей процес. В результаті ми отримуємо недорогу платформу, яка підходить для розробки, складання прототипів та виробництва.

4.2.4 Підключення контактів та живлення

Необхідно мати пристрій ESP8266 NodeMCU та чіп ATECC508A. ATECC508A можна отримати або як плату ATCRYPTOAUTH-XPRO, яка не вимагає пайки, або голі контакти ATECC508A, що вимагає пайки (рис. 4.5).

Таблиця 4.1 – Опис контактів плат необхідних для реалізації макету

Функція	ATECC508A pin	ESP8266 pin	pin NodeMCU	ATCRYPTOAUTH pin
SDA	5	10 (GPIO12)	D6	11 (yellow)
SCL	6	9 (GPIO14)	D5	12 (white)
GND	4	Any suitable	GND	19 (black)
VCC	8	Any suitable	3V3	20 (red)

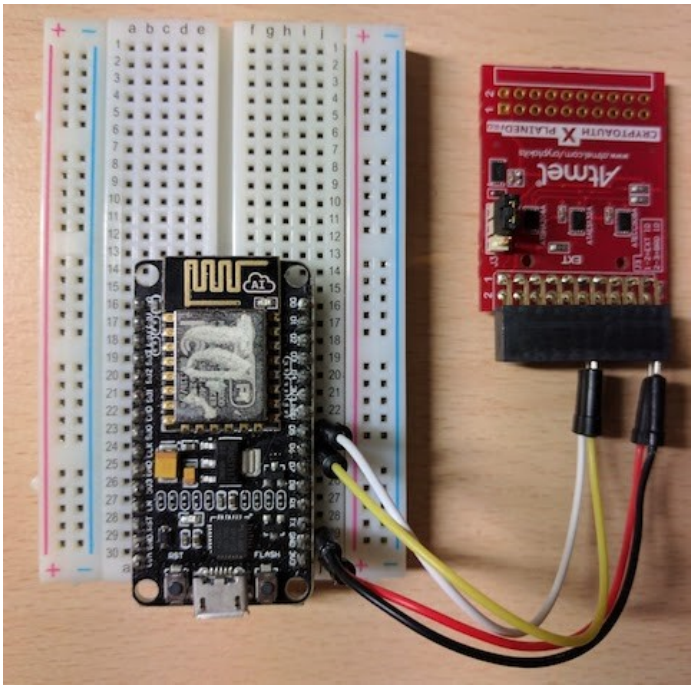

Рисунок 4.5 – Підключення до ATCRYPTOAUTH-XPRO

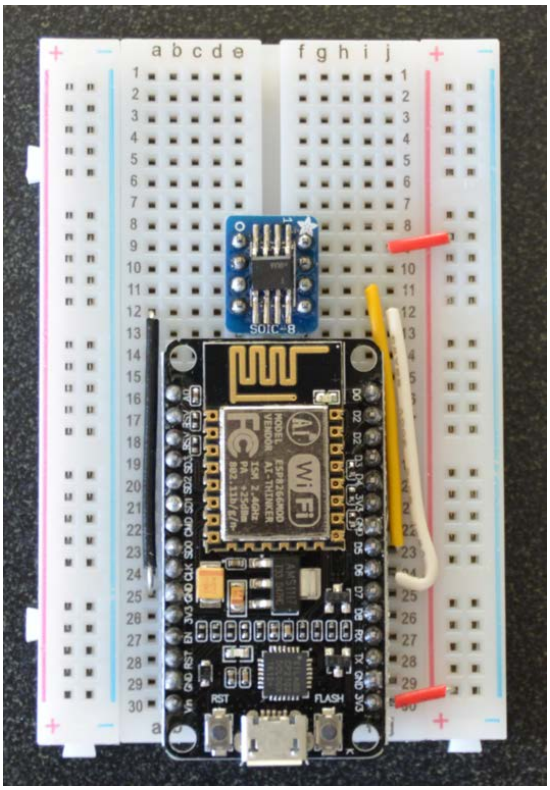

Рисунок 4.6 – підключення контактів ATECC508A

4.3 Налаштування мережних зв’язків

Коли мікросхема ATECC508A підключена, настав час конфігурації пристрою.

1. Створюємо сертифікат і ключ. Ви можете створити самопідписаний сертифікат або скористатися власною службою сертифікації (CA). Потрібно буде генерувати сертифікат алгоритму цифрового підпису еліптичної кривої (ECDSA) за допомогою кривої P-256, оскільки ATECC508A (рис. 4.6) підтримує цей тип сертифіката.

```
$ openssl ecparam -out ecc.key.pem -name prime256v1 -genkey
$ openssl req -new -subj \
 "/C=IE/L=Dublin/O=ACME Ltd/OU=Testing/CN=test.acme.com" \
 -sha256 -key ecc.key.pem -text -out ecc.csr.tmpl
$ openssl x509 -in ecc.csr.tmpl -text -out ecc.crt.pem \
 -req -signkey ecc.key.pem -days 3650
```

2. Прошиваємо свій пристрій за допомогою Mongoose OS, як було описано в кроці 1.

3. Використовуємо функцію Mongoose OS I2C.Scan, щоб перевірити, чи мікосхема правильно підключена та працює, як очікувалося. Очікується, що інструмент mos відповість на [96], що є адресою I2C ATECC508A. Якщо цього немає, потрібно повернутися та перевірити підключення або спробувати іншу мікросхему, якщо це можливо.

```
$ mos call I2C.Scan
[ 96 ]
```

4. Налаштовуємо мікросхему. Ми можемо використовувати зразок конфігурації, наданий у сховищі Mongoose OS Git. Збережемо конфігурацію як «atca-aws-test.yaml» та встановлюємо її за допомогою розширених команд «mos»:

```
$ mos config-set sys.atca.enable=true
$ mos -X atca-set-config atca-aws-test.yaml --dry-run=false
$ mos -X atca-lock-zone config --dry-run=false
$ mos -X atca-lock-zone data --dry-run=false
```

Ці зміни незворотні: щойно зони заблоковані, їх неможливо розблокувати. Крім того, ця вибіркова конфігурація захищена і підходить лише для тестування і ні в якому разі не може використовуватися для виробничих інтеграцій. При створенні виробничої конфігурації потрібно звернутися до посібника Microchip та іншої документації.

5. Записуємо створений ключ у захищений елемент. Враховуючи те, що ми використовували конфігурацію зразка, це двоступеневий процес:

а. Створюємо та встановлюємо ключ шифрування у слот 4:

```
$ openssl rand -hex 32 > slot4.key
$ mos -X atca-set-key 4 slot4.key --dry-run=false
```

```
AECC508A rev 0x5000 S/N 0x012352aad1bbf378ee, config is locked, data is locked
Slot 4 is a non-ECC private key slot
SetKey successful.
```

б. Встановлюємо ключ ECC у слот 0:

```
$ mos -X atca-set-key 0 ecc.key.pem --write-key=slot4.key --dry-run=false
```

```
AECC508A rev 0x5000 S/N 0x012352aad1bbf378ee, config is locked, data is locked
Slot 0 is a ECC private key slot
Parsed EC PRIVATE KEY
Data zone is locked, will perform encrypted write using slot 4 using slot4.key
SetKey successful.
```

7. Прописуємо конфігурацію сервера HTTP, щоб використовувати завантажений сертифікат та приватний ключ із слота пристрою 0:

```
$ mos config-set http.listen_addr=:443 http.ssl_cert=ecc.crt.pem http.ssl_key=ATCA:0

Getting configuration...
Setting new configuration...
Saving and rebooting…
```

Під час запуску ми повинні побачити наступне в журналі пристрою:

```
mgos_sys_config_init_http HTTP server started on [443] (SSL)
```

А під час підключення до браузера слід побачити таке:

```
ATCA:2 ECDH get pubkey ok
ATCA:0 ECDSA sign ok
ATCA:2 ECDH ok
```

Виконуємо налаштування AWS IoT та підключаємося дотримуючись прикладу MQTT у сховищі Mongoose OS Git. Після встановлення облікових даних Wi-Fi, запустимо цю команду, щоб забезпечити плату ESP8266 в AWS IoT та використовувати захищений елемент:

```
$ mos aws-iot-setup --use-atca --aws-iot-policy=mos-default
```

У цей момент слід підключитися до AWS IoT за допомогою захищеного елемента. На ESP8266 час узгодження з'єднання буде знижено з 10 секунд або більше до менш ніж однієї секунди. Сертифікат захищений в ATECC508A і буде використовуватися для автентифікації вашого пристрою на AWS IoT, коли потрібно

повторно підключитися. Ця платформа зробить нас на крок ближче до безпечного розгортання виробництва.

Висновки до розділу

В даному розділі розглянуті потенційні можливості роботи з мережними сервісами при розробці та інтеграції пристроїв Інтернету речей з прикладами сучасних методів захисту.

Наш світ стрімко змінився, зараз ми можемо реалізовувати свої ідеї на значно більш високому рівні. І тут, дуже доречно йти нога в ногу з технологіями і звернути належну увагу на мережні сервіси IoT від Amazon Web Services, які надають такі переваги в плані забезпечення безпеки:

– виконання автентифікації за допомогою сертифіката X.509 через з’єднання MQTT;

–для створення сертифіката X.509, необхідно створити запит на підписання сертифіката (CSR) та надати його органу сертифікації;

–використання протоколу TLS 1.2, який є спадкоємцем протоколу SSL, який аналогічний тому, що використовується для безпечного здійснення покупок або в інтернет банкінгу, але крім автентифікації сервера, клієнт також використовує сертифікат X.509 для підтвердження своєї ідентичності;

–щоб завершити процес автентифікації, AWS IoT обчислює хеш для всіх записів зв'язку, які є частиною поточного сеансу з сервером AWS IoT. Потім він обчислює цифровий підпис для цього хеша, використовуючи його приватний ключ;

–для надання дозволів, необхідно додати до сертифіката політику так само, як користувачеві IAM. Політика за замовчуванням надає власнику сертифіката права лише на публікацію теми, визначеної в атрибуті "Ресурс".

На сьогодні AWS IoT пропонує комплексний підхід до вирішення проблеми безпеки, що вже зараз можна побачити на прикладі сучасних хмарних обчислень.

5 СТАРТАП-ПРОЕКТ

5.1 Основні відомості

Сутність стартап-проекту. Досліджуючи ринок Інтернету речей було виявлено можливість вдосконалення безпеки систем, технологічних комплексів та одиничних включень, які впроваджуються щодня все з більшими темпами та в більших масштабах за допомогою використання хмарних технологій. Зміст ідеї стартапу та визначення її характеристик наведено в табл. 5.1 та табл. 5.2.

Таблиця 5.1 – Зміст ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Запропонувати ефективне, гнучке та масштабоване рішення для підвищ щеня надійності та забезпечення безпеки в системах IoT в умовах різних проектів.	1. Промисловість	Управління великою кількістю кінцевих пристроїв
	2. Повсякденне життя	Захист персональної інформації та безпека операторів та користувачів
	3. Вся електроніка, яка має вихід в мережу Інтернет	Гнучка, масштабована, технологічна перевага даної реалізації

Таблиця 5.2 – Визначення характеристик ідеї стартап-проекту

№ п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів		W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Запропонований метод	Загальновживаний метод			
1.	Інтеграція систем різного рівня складності з використанням хмарних рішень в галузі безпеки в залежності від вимог замовника	Дає змогу	Дає змогу	Залежить від конретних випадків застосування, від обладнання	Цінова політика може не задовільнити кінцевого споживача	Мінімальний набір обладнання та кроків при інтеграції
2.	Висока відмово стійкість, доступність та постійне оновлення відповідно до актуальних вразливостей	Дає змогу	Не дає змогу	Немає 100-відсоткового захисту	Потрібна підтримка для моніторинг для більшої ефективності	Раціональне співвідношення ціни та якості, за помірну плату, споживач отримує передові технології

5.2 Технологічний аудит ідеї стартап-проекту

У таблиці 5.3 оцінено можливість технологічної реалізації ідеї стартапу та показано технології, які можна застосувати для реалізації проекту.

Таблиця 5.3. Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Інтеграція рішень безпеки для систем Інтернету речей в рамках різних проектів та компаній які працюють з мережевими технологіями та IoT	Спеціалізоване обладнання на базі існуючих IoT пристроїв	Присутня	Доступна в випадку достатнього бюджету
2		Використання апаратних систем, вузького призначення розроблених на мікроконтролерах	Необхідно розробити	Доступна в випадку достатнього бюджету
3		Програмні рішення та хмарні технології на всіх вузлах мережі	Присутня	Доступна в випадку достатнього бюджету

Обрана технологія реалізації ідеї проекту: застосування комплексу програмних рішень та хмарних сервісів для оптимізації систем інтернету речей та підвищення рівня безпеки.

5.3 Аналіз можливостей ринку для запуску проекту

У таблиці 5.4 показано попередню характеристику потенційного ринку стартап-проекту.

Таблиця 5.4. Попередня характеристика потенційного ринку стартапу

№ п/п	Показники ринку (найменування)	Характеристика
1	Кількість основних гравців, од	2
2	Обсяг продажів, грн/ум.од	950000
3	Тенденції ринку (якісна оцінка)	Швидко зростає
4	Обмеження для входу (вказати характер обмежень)	Пошук потенційних клієнтів
5	Специфічні вимоги стандартизування та сертифікування	Ліцензія на діяльність, сертифікація від компаній вендорів, сертифікація співробітників в галузі мережевих технологій
6	Середня норма рентабельності в даній галузі, %	950000/610000 = 156%

У таблиці 5.5 показано характеристику потенційних клієнтів стартап-проекту

Таблиця 5.5. Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності поведінки потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Реалізація політик безпеки при розгортанні технічних систем	Компанії, які мають технічну складову	Необхідний рівень безпеки відповідно до типу компанії	Результат повинен відповідати найвищим стандартам безпеки актуальним відповідним загрозам

				та вразливостям які змінюються з кожним днем
2	Відповідальність, збитки та наслідки у випадку вразливості електронних систем	Компанії, які працюють з персональними даними та іншими джерелами цінної інформації, або контролюють електронні системи які несуть техногенну загрозу	Кожна група має компаній має власні вимоги до технічного забезпечення та політик і засобів безпеки відповідно	Забезпечення безпеки в залежності від потреб споживача

У табл. 5.6 наведено основні загрози реалізації стартап-проекту.

Таблиця 5.6. Фактори загроз

№ п/п	Фактор	Опис загрози	Планове реагування компанії
1	Конкуренція	Велика кількість пристроїв та засобів безпеки на сьогодні	Реалізація послуг на найвищому рівні та на обладнанні провідних вендорів для надання максимально можливих та гнучких послуг відповідно до потреби клієнта

2	Швидка зміна ринку та технологій	Складність відповідати тенденціям ринку безпеки для надання актуальних послуг	Інвестиції в сертифікацію співробітників, моніторинг сучасних рішень від вендорів, які враховують потреби так званого «завтрашнього дня»

У табл.5.7 наведено основні можливості під час реалізації стартап-проекту.

Таблиця 5.7. Основні можливості

№ п/п	Фактор	Опис можливості	Планове реагування компанії
1	Лідерські позиції на ринку інтеграції рішень безпеки	Стрімке зростання попиту та кількісне зростання IoT систем	Якісне та кількісне збільшення потужностей
2	Впровадження запропонованих технологій в уже існуючі системи IoT	Збільшення об’ємів закупівель та пошук технологічних рішень для ширшого охоплення ринку	Якісне та кількісне збільшення потужностей

У таблиці 5.8 наведено особливості та вплив конкурентного середовища на впровадження проекту.

Таблиця 5.8. Аналіз конкуренції

Особливості конкурентного середовища	Прояв даної характеристика	Вплив на діяльність підприємства (планові дії компанії для забезпечення конкурентоспроможності)

1.Конкуренція	Застосування вже існуючих технологій	Проведення стандартизації на високому рівні
2.Локальний	Відсутність єдиного постачальника послуг	Індивідуальний підхід до кожної локальної ділянки
3.Міжгалузева	Відсутня	Відсутня
4.Товарно-видова	Використання стандартизованих технологій	Застосування загальновживаних апаратних та програмних засобів, за необхідності
5.Цінова	Використання високовартісних спеціалізованих комплексів	Використання гнучких універсальних програмних засобів для компенсації апаратної частини
6.Марочна	Кожна діагностика повинна бути стандартизованою	Здобуття лідерських позицій на ринку інтеграції рішень безпеки для систем IoT

У таблиці 5.9 проаналізовано конкуренцію проекта в галузі за М. Портером

Таблиця 5.9. Аналіз конкуренції за М. Портером

Складові аналізу	Прямі конкуренти	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Системні інтегратори	Нові гравці ринку систем IoT	Залучення лише провідних в галузі вендорів та співпраця з хмарними сервісами для забезпечення охоплення ринку та низовий ціновий сегмент	Постійно зростаюча категорія в умовах сучасного всеохоплюючого Інтернету речей	Надання переваги компаніям які займають лідерські позиції та мають репутацію інноваційних та технологічних протягом всього часу існування на ринку
Висновки:	Середня	Є можливість виходу на ринок	Постачальники встановлюють цінову	Клієнти встановлюють вимоги до якості	Обмеженнь немає

			політику на обладнання		

У табл. 5.10 наведено та обґрунтувано фактори конкурентноспроможності.

Таблиця 5.10. Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (чинники, що роблять фактор порівняння конкурентних проектів значущим)
1	Раціональніша цінова політика	Можливість раціональнішого використання ресурсів
2	Забезпечення дистриб'юторських послуг	Постачання та інтеграція систем як «під ключ» так і окремих компонентів відповідно до аудиту технічної інфраструктури замовника та поданих ним ТЗ

У табл. 5.11 перелічено сильні та слабкі сторони проекту.

Таблиця 5.11. Порівняльний аналіз сильних та слабких сторін проекту

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Порівняння рейтингу товарів-конкурентів						
			–3	–2	–1	0	+1	+2	+3
1	Раціональніша цінова політика	15			+				
2	Дистриб'юція на різних етапах	16				+			
3	Пост-продажна підтримка	10				+			
4	Потреба в залученні висококваліфікованих кадрів	19	+						

У табл.5.12 представлений SWOT-аналіз стартап-проекту.

Таблиця 5.12. SWOT- аналіз стартап-проекту

Сильні сторони: раціональна цінова політика, постачання апаратного та програмного забезпечення	Слабкі сторони: потреба в залученні висококваліфікованих кадрів, постійна перепідготовка та актуалізація знань, стеків протоколів та програмних і апаратних засобів
Можливості: Інноваційна технологічно-економічна модель інтеграціх хмарних сервісів для реалізації безпеки на різних рівнях та в різних елементах технічних систем IoT	Загрози: Конкуренція на швидко зростаючому та ринку, його збільшення, нові гравці та технології

Альтернативи ринкового впровадження стартапу показані в табл.5.13.

Таблиця 5.13. Альтернативи ринкового впровадження роекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність залучення ресурсів	Терміни реалізації
1	Складання договорів технічними компаніями для інтеграції та постачання на ексклюзивних правах в залежності від обсягів замовлення для подавлення конкурентів замовником за рахунок реалізованих системних рішень	висока	короткі
2	Застосування інноваційних технологій та комплексів систем і програмного забезпечення для швидкого зростання на ринку	висока	короткі

5.4. Розроблення ринкової стратегії проекту

Обгрунтування вибору цільових груп потенціійних споживачів показано в табл. 5.14 [31].

Таблиця 5.14. Вибір цільових груп потенційних споживачів

№ п/п	Загальний профіль цільової групи	Готовність сприйняття продукту споживачами	Орієнтовний попит цільової групи (сегменту)	Напруженість конкуренції в сегменті	Складність входу у сегмент

	потенційних клієнтів				
1	Компанії з технічною інфраструктурою, з високим рівнем безпеки	Висока	Високий	Середня	Середня
2	Системи Інтернету речей в різних галузях та напрямках які на сьогодні широко розповсюджені	Середня	Середній	Середня	Низька

Визначення базової стратегії розвитку наведено у табл. 5.15.

Таблиця 5.15. Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Основні конкурентоспроможні позиції згідно з обраною альтернативою	Базова стратегія розвитку*
1	Дистриб'юція окремих елементів	Впровадження нового стандарту якості та клієнтоорієнтованості	Залучення ключових гравців у сфері телекомунікаційних систем	Стратегія диференціації
2	Бюджетність проекту в порівнянні з іншими гравцями ринку	Інвестиція в кваліфіковані кадри	Використання унікальних, інноваційних, передових рішень для досягнення лідерських позицій	Стратегія лідерства по якості послуг та рівню обслуговування

Визначення основної стратегії конкурентної поведінки показано в табл. 5.16.

Таблиця 5.16. Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект унікальним на ринку?	Чи необхідно буде компанії шукати нових споживачів, чи опрацьовувати існуючих у конкурентів?	Чи необхідно компанії копіювати основні характеристики товару конкурента?	Стратегія конкурентної поведінки*
1	Так	Опрацьовувати існуючих та шукати нових	Немає необхідності	Стратегія інноваційної конкуренції

Визначення стратегії позиціонування показано в табл. 5.17.

Таблиця 5.17. Визначення стратегії позиціонування

№ п/п	Вимоги цільової аудиторії до товару	Основна стратегія розвитку	Основні конкурентоспроможні позиції стартап-проекту	Визначення асоціацій, які сформують комплексну позицію стартап-проекту (три основних)
1	Належна висока якість послуг	Стратегія диференціації	Новизна, гарант якості, точність дослідження	Якість, точність, надійність
2	Раціональні витрати	Стратегія лідерства по витратах	Гнучкість запропонованого рішення	Універсальність, інноваційність, надійність

5.5. Розроблення маркетингової програми стартап-проекту

Основні переваги концепції потенційного товару показано в табл. 5.18.

Таблиця 5.18. Визначення основних переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Основні переваги перед конкурентами (існуючі або потенційні)
1	Якість	Належна висока якість, надійність	Масштабованість, гнучкість, якість
2	Раціональна вартість	Оптимальне використання коштів, максимальна якість обладнання від провідних вендорів, максимальний рівень кваліфікації спеціалістів в залежності від вартості та складності проекту	Раціоналізація витрат відповідно до розміру бюджету замовника

Виявлено три рівні моделі товару. Зміст та складові рівнів товару показано в табл. 5.19.

Таблиця 5.19. Опис трьох рівнів моделі товару

Рівні товару	Зміст та складові		
I. Товар за задумом	Якісний товар та послуги, стандартизована якість послуг та обладнання		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1)Вартість обслуговування, 2)Кількість комплектів обладнання 3)Строк безвідмовної експлуатації 4)Технологічна собівартість товару	1) М 2) М 3) М 4) М	1)Е 2) Пр 3)Нд 4)Тх
	Якість: міжнародні стандарти, постійне обслуговування та підтримка обладнання		
	Постачання, розрахунки та інтеграція під конкретні системи		
	Марка: Системи безпеки		
III. Товар із підкріпленням	До продажу – обладнання та встановлення		
	Після продажу – аудит та вдосконалення застарілих елементів та систем в цілому в залежності від актуальних вимог та потреб		

Потенційний товар буде захищено від копіювання завдяки: товарна марка та унікальні рішення, які не мають аналогів на ринку та відрізняються між собою оскільки кожне з рішень є глибоко індивідуальним в залежності від потреб замовника, що необхідно для забезпечення найвищих та актуальних на майбутнє стандартів безпеки.

Визначення цінової політики на послугу показано в табл. 5.20.

Таблиця 5.20. Визначення меж встановлення ціни

№ п/п	Цінова політика товарів-замінників	Цінова політика на товари-аналоги	Рівень купівельної спроможності цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	40000 у.о./од. (стандартні системи безпеки)	-	Дуже високий	Н.1000 у.о. – В.100000 у.о. (Товар) Н.1000 у.о. –

				В.100000 у.о. (Послуга)

Створення системи збуту послуги вказано у табл. 5.21.

Таблиця 5.21. Створення системи збуту

№ п/п	Закупівельна поведінка цільових клієнтів	Функції збуту, що повинен забезпечувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Орієнтована на максимальний рівень безпеки в системах різного роду	Поставки якісного обладнання та інноваційних рішень	Значна	Контрактна система

Концепції маркетингових комунікацій показано в табл. 5.22.

Таблиця 5.22. Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій цільових клієнтів	Основні методи позиціонування	Завдання рекламного звернення	Концепція рекламного звернення
1	Зацікавленість якісному та якісному продукті з раціональним використанням ресурсів	Мережеві ресурси	Гарантія якості та стандартизація, сервісна політика	Привернути увагу до покращень, пов’язаних із зростаючою потребою в захисті	Позиціонування безпеки як основи для побудови надійних рішень та іміджу компанії
2	Зацікавленість у великих об’ємах продукції із дотриманням умов якості	Мережні ресурси	Глибина каналу постачальників, гарант якості	Привернути увагу до переваг над іншими гравцями ринку	Позиціонування як активного та інноваційного гравця ринку

					на фоні конкурентів

Висновки до розділу

1. Виявлено, що комерціалізацію стартап-проекту щодо застосування та розвитку комплексу апаратних та програмних рішень для забезпечення безпеки в системах Інтернету речей можна вважати доцільною та актуальною в умовах надшвидкого розвитку галузі. На ринку технічних рішень попит на дану пропизицію все більше набирає опитів та зростає у обсягах, який зараз задовільняють товари замінники та більш вузьконаправлені рішення, саме тому необхідно виходити на ринок та пропонувати широкий спектр рішень для забезпечення потреб ринку та розвитку спектру послуг. Рентабельність на ринку забезпечить в першу чергу можливість заміни існуючих рішень на більш масштабовані, гнучкі та інноваційні, шляхом застосування не компексних рішень.

2. Перспективність впровадження досить висока, адже основними клієнтами є компанії, які активно впроваджують системи Інтернету речей, які активно освоюють ринок, і, в разі досягенення відповідного авторитету, існує можливість охоплення у масштабов міжнаціональних ринків. Конкурентноспроможність проекту забезпечує високий рівень кваліфікації у підході до вирішення та реалізації кожного конкретного рішення та звернення зі сторонни клієнта, максимальна гнучкість у плані рівня реалізованих систем в залежності від типу, бюджету клієнта та особливостей застосування.

3. Обрана альтернатива впровадження – пошук актуальних та інноваційних рішень, їх дистриб’юція та популяризація в умовах ринку. Імплементація проекту

доцільна, а сприятливі умови для його розвитку обумовлені рентабельністю та зацікавленістю потенційних груп компаній та окремих клієнтів.

ВИСНОВКИ

У магістерській дисертації вирішено проблему забезпечення безпеки в системах Інтернету речей в процесі інтеграції та розгортання систем такого типу в різноманітних умовах, що забезпечить гнучкість, масштабованість та технологічність таких рішень в умовах стрімкого розвитку технологій IoT та відповідно нових атак та вразливостей.

1. Проаналізовано основні вразливості систем Інтернету речей та типи атак, які існують на сьогодні, наслідки яких можуть бути спричинені як для підприємств так і для кінцевих користувачів.

До основних проблем безпеки віднесено атаки на сервери, робочі станції та смартфони, слабку автентифікацією, незашифровані повідомлення, що надсилаються між пристроями, бази даних SQL та відсутність належного контролю оновлення програмного забезпечення та політик безпеки.

Проте, у випадку ретельного розуміння ключових моментів безпеки, інтеграцію систем захисту може бути виконано із забезпеченням відповідних політик безпеки в системах та пристроях на всьому підприємстві. На сьогодні AWS IoT пропонує комплексний підхід до вирішення проблем безпеки, що вже зараз можна побачити на прикладі сучасних мережних розподілених обчислень.

2. Досліджено протоколи, які використовують для встановлення захищеного з’єднання, передавання повідомлень та автентифікації. Найпоширенішим протоколом для передавання повідомлень залишається MQTT-протокол, проте для забезпечення безпеки його рекомендовано використовувати в комплексі з протоколом TLS.

Розкрито переваги використання протоколу TLS 1.2, який аналогічний протоколу, що використовується для безпечного здійснення покупок або в інтернет банкінгу, але крім автентифікації сервера клієнт також використовує сертифікат X.509 для підтвердження своєї ідентичності.

3. Проаналізовано експлуатаційні обмеження в залежності від типу та умов використання, які часто не дозволяють безпосередньо використовувати основні

заходи безпеки, такі як реалізація брандмауерів або використання криптосистем для шифрування зв'язку з іншими пристроями, та вплив низької ціни та вузьконаправленості багатьох пристроїв на те, що виробники дуже рідко використовують надійні системи захисту.

4. Досліджено та проаналізовано пропозиції щодо безпеки, які надають веб-служби Amazon Web Services (AWS), компоненти, запропоновані Cisco (Fog Computing), а також Microsoft Azure, які можуть бути ефективно інтегровані в системи Інтернету речей шляхом вибору виробника в залежності від потреб замовника.

5. Отримано сертифікат для плати ESP8266, який зареєстровано на мережній платформі AWS IoT, що зберігається на локальному комп’ютері клієнта. Створено образ вбудованого ПЗ для ESP8266, яке може підключатися безпосередньо до AWS IoT і бути налаштоване за допомогою кінцевої точки AWS IoT, пароля Wi-Fi SSID, не змінюючи його середовище побудови, яке можна використовувати для розроблення нових мікропрограм. Використано плату криптографічного захисту ATECC508A для підвищення безпеки та продуктивності системи. На основі мікроконтролерів, що використовують ATECC508A, можна встановити TLS-з'єднання швидше, ніж лише на базі програмного забезпечення.

6. Розроблено стартап-проект, який базується на просуванні на ринок Інтернету речей рішень на базі досліджуваного макету обладнання як комплексного рішення для інтеграції в системи IoT на різних етапах життєвого циклу. Проведено дослідження доцільності та рентабельності даного бізнес-проекту та визначено, що комерціалізація проекту є доцільною.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Технічний огляд системи Cisco IOX URL: https://developer.cisco.com/site/iox/technical-overview/

2. Опис системи AWS IoT URL: http://aws.amazon.com/iot/

3. Платформа Інтернету речей від IBM URL: https://docs.internetofthings.ibmcloud.com/devices/mqtt.html

4. Короткий огляд перспектив 5G мереж URL: http://www.techrepublic.com/article/does-the-world-really-need-5g/

5. Хмарне сховище Amazon S3 Glacier URL: https://aws.amazon.com/glacier/

6. Технічні специфікації AWS IoT URL: https://docs.aws.amazon.com/iot/latest/developerguide/thingshadow-mqtt.html

7. Cisco Smart розумний паркінг, опис реалізації URL: https://www.cisco.com/web/strategy/docs/parking_aag_final.pdf

8. Вразливості стандарту ZigBee URL: https://www.blackhat. com/docs/us-15/materials/us-15-Zillner-ZigBee-Exploited-The-Good-The- Bad-And-The-Ugly-wp.pdf

9. Використання дронів для пошуку IoT пристроїв URL: http://fortune.com/2015/08/05/researchers-drone-discover-connected-devices-austin/

10. Опис системи SP800-64 URL: http://csrc.nist.gov/publications/nistpubs/800-64-Rev2/SP800- 64-Revision2.pdf

11. Вплив систем Інтернету речей на дата центри майбутнього URL: http://www.forbes.com/sites/moorinsights/2015/08/04/how-theinternet-of-things-will-shape-the-datacenter-of-the future

12. Опис особливостей застосування туманних обчислень Cisco URL: http://blogs.cisco.com/perspectives/iot-from-cloud-to-fog-computing

13. Огляд системи OpenDNS URL: https://www.opendns.com/enterprisesecurity/

14. Специфікація платформи Azure URL: https:// azure.microsoft.com/en-us/documentation/articles/iot-hub-devguide/

15. Опис хмарного IoT проекту URL: http://www.easures.com/document.asp?doc_id = 1325828

16. Посібник з безпеки Bluetooth URL: https://www.nsa.gov/ia/_files/ i732-016r-07.pdf

17. Посібник з безпеки Bluetooth по NIST SP 800-121 URL: http://csrc.nist.gov/nistpubs / 800-121-rev1 / sp800-121_rev1.pdf

18. Brian Russell, Practical Internet of Things Security, Packt Publishing Ltd, 2016

19. Guenter Schaefer, Michael Rossberg, Security in Fixed and Wireless Networks, John Wiley & Sons Ltd, 2014

20. Xiangyun Zhou, Lingyang Song, Yan Zhang, Security in Fixed and Wireless Networks, Taylor & Francis Group, LLC, 2014

Додаток А

РЕФЕРАТ

англійською мовою за темою магістерської дисертації

ABSTRACT

Only a few people would contest the assertion that the phenomenon of the Internet of Things poses problems related to security, safety, and privacy. Given the remarkable industrial and consumer diversity of the IoT, one of the principal challenges and goals we faced when electing to write this book was determining how to identify and distill the core IoT security principles in as useful, but industry-agnostic a way as possible. It was equally important to balance real-world application with background theory, especially given the unfathomable number of current and forthcoming IoT products, systems, and applications. To end this, we included some basic security (and safety) topics that we must adequately, if minimally, cover as they are needed as a reference point in any meaningful security conversation. Some of the security topics apply to devices (endpoints), some to communication connections between them, and yet others to the larger enterprise.

Another goal of this book was to lay out security guidance in a way that did not regurgitate the vast amounts of existing cybersecurity knowledge as it applies to today's networks, hosts, operating systems, software, and so on, though we realized some is necessary for a meaningful discussion on IoT security. Not wanting to align with a single industry or company selling products, we strove to sufficiently carve out and tailor useful security approaches that encompass the peculiarities and nuances of what we think both distinguishes and aligns IoT with conventional cybersecurity. A wide range of both legacy industries (for example, home appliance makers, toy manufacturers, automotive, and so on) and startup technology companies are today creating and selling connected devices and services at a phenomenal and growing rate. Unfortunately, not all are terribly secure—a fact that some security researchers have unrelentingly pointed out, often with a sense of genuine concern. Though much of the criticism is valid and warranted, some of it has unfortunately been conveyed with a certain degree of unhelpful hubris.

Interestingly, however, is how advanced some of the legacy industries are with regard to high-assurance safety and fault-tolerant design. These industries make extensive use of the core engineering disciplines—mechanical, electrical, industrial, aerospace, and control

engineering—and high-assurance safety design to engineer products and complex systems that are, well, pretty safe. Many cybersecurity engineers are frankly ignorant of these disciplines and their remarkable contributions to safety and fault-tolerant design. Hence, we arrive at one of the serious obstructions that IoT imposes to achieving its security goals: poor collaboration between safety, functional, and security engineering disciplines needed to design and deploy what we term **cyber-physical systems** (**CPS**). CPS put the physical and digital engineering disciplines together in ways that are seldom addressed in academic curricula or corporate engineering offices. It is our hope that engineers, security engineers, and all types of technology managers learn to better collaborate on the required safety and security-assurance goals.

While we benefit from the IoT, we must prevent, to the highest possible degree, our current and future IoT from harming us; and to do this, we need to secure it properly and safely. We hope you enjoy this book and find the information useful for securing your IoT.

One struggles to find it covered in academic curricula outside of a few university computer science programs, network engineering, or dedicated security programs such as SANS. Most security practitioners have strong computer science and networking skills but are less versed in the physical and safety engineering disciplines covered by core engineering curricula. So, the cyber-physical aspects of the IoT face a safety versus security clash of cultures and conundrums:

– everyone is responsible for security;

– the IoT and CPS expose huge security problems crisscrossing information computing and the physical world;

– most traditional, core engineering disciplines rarely address security engineering (though some address safety);

– many security engineers are ignorant of core engineering disciplines (for example, mechanical, chemical, electrical), including fault-tolerant safety design.

Because the IoT is concerned with connecting physically engineered and manufactured objects — and thus may be a CPS — this conundrum more than any other comes into play. The IoT device engineer may be well versed in safety issues, but not fully

understand the security implications of design decisions. Likewise, skilled security engineers may not understand the physical engineering nuances of a *thing* to ascertain and characterize its physical-world interactions (in its intended environment) and fix them. In other words, core engineering disciplines typically focus on functional design, creating things to do what we want them to do. Security engineering shifts the view to consider what the thing can do and how one might misuse it in ways the original designer never considered. Malicious hackers depend on this. The refrigeration system engineer never had to consider a cryptographic access control scheme in what was historically a basic thermodynamic system design. Now, designers of connected refrigerators do, because malicious hackers will look for unauthenticated data originating from the refrigerator or attempt to exploit it and pivot to additional nodes in a home network.